IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DATA MINING FOR INTERNET OF THINGS

Received January 13, 2020, accepted January 19, 2020, date of publication February 6, 2020, date of current version May 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2972026

A Blockchain-Based Application System for Product Anti-Counterfeiting

JINHUA MA[1], SHIH-YA LIN[2], XIN CHEN[1], HUNG-MIN SUN[2], YEH-CHENG CHEN[3], (Graduate Student Member, IEEE) AND HUAXIONG WANG[4]

[1]Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China
[2]Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan
[3]Department of Computer Science, University of California at Davis, Davis, CA 95616, USA
[4]Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 639798

Corresponding authors: Xin Chen (xinchen9503@outlook.com) and Hung-Min Sun (hmsun@cs.nthu.edu.tw)

The work of Jinhua Ma and Xin Chen was supported by the National Natural Science Foundation of China under Grant 61872089, Grant 61902070, and Grant 61972094. The work of Shih-Ya Lin and Hung-Min Sun was supported by the Ministry of Science and Technology of the Republic of China under Grant MOST 106-2221-E-007-026-MY3, Grant MOST 107-2221-E-007-015-MY3, and Grant MOST 108-2218E001-001. The work of Huaxiong Wang was supported by the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative and Singapore Ministry of Education under Research Grant MOE2016-T2-2-014(S).

ABSTRACT In recent years, blockchain has received increasing attention and numerous applications have emerged from this technology. A renowned Blockchain application is the cryptocurrency Bitcoin, that has not only been effectively solving the double-spending problem but also it can confirm the legitimacy of transactional records without relying on a centralized system to do so. Therefore, any application using Blockchain technology as the base architecture ensures that the contents of its data are tamper-proof. This paper uses the decentralized Blockchain technology approach to ensure that consumers do not fully rely on the merchants to determine if products are genuine. We describe a decentralized Blockchain system with products anti-counterfeiting, in that way manufacturers can use this system to provide genuine products without having to manage direct-operated stores, which can significantly reduce the cost of product quality assurance.

INDEX TERMS Blockchain, ethereum, counterfeit.

I. INTRODUCTION

A. MOTIVATION

The trade in counterfeit goods is growing and is affecting the sales and profits of companies affected by this phenomenon. To ensure the identification and traceability of real products throughout the supply chain, this paper is the first to propose a fully functional blockchain system to prevent product counterfeiting. Enterprises only need to pay very low transaction fees, and they no longer need to worry about the possibility of obtaining counterfeit products.

The related anti-counterfeiting technology has already been proposed but not yet perfected. For example in ''Makerchain: A blockchain with chemical signature for self-organizing process in social manufacturing'' [1] presents an anti-counterfeiting method composed of chemical signature to represent unique features of personalized products, and in ''Block-Supply Chain: A New Anti-Counterfeiting Supply Chain Using NFC and Blockchain'' [2] proposed a new decentralized supply chain (block-supply) utilizing blockchain and NFC technologies. But All of these methods still cannot prevent seller selling counterfeit products to customer.

Nowadays, small and medium-sized enterprises (SMEs) often have financial burdens, which cannot be compared with large companies with strong financial resources. In the brand management sector, SMEs will inevitably need to reduce costs and will be most likely unable to implement traditional methods of preventing counterfeited goods.

In this paper, we propose to implement a Blockchain architecture provided by Ethereum to record product ownership on the Blockchain. By using Blockchain's untraceability and transparency properties, and the assurance that each record cannot be forged on the Blockchain, consumers don't need to fully rely on trusted third parties to safely know the source of the purchased product. SMEs can

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

implement the anti-counterfeit application system proposed in this paper, and they will only need to pay a relatively low-cost for the operation fees to achieve secure and unforgeable anti-counterfeit authentication.

B. CONTRIBUTION

The companies using this system can increase users trust in the brand. Solving the problem of SMEs, that are unable to open directly-operated stores and could not cooperate with large-scale chain dealers.

In an overview of our system, it is aimed to solve the problem of brand anti-counterfeiting certification, focusing in expanding sales channels, and providing to small vendors the chance to prove the source of each component of their product.

The system is set up on a Blockchain, and companies that implement this system will only need to pay an amount of money needed to create and change their contract status. Using fully disclosed smart contract information, anyone can easily prove the legitimate source of the business and can also serve as proof for the consumer's purchase of goods. For retailers, it is possible to prove whether they provide genuine goods by using this anti-counterfeit Blockchain system and no longer have to be concerned about competing with counterfeits sold at low prices.

C. ORGANIZATION

The rest of this paper is organized as follows: In Chapter II, we examine existing related work. In Chapter III, we introduce our system design pattern including process performance. In Chapter IV, we describe the implementation of our system. In Chapter V, we show the result of time requirements and money demand in implementing our system. In Chapter VI, we provide conclusion and discuss future work.

II. BACKGROUND

A. BLOCKCHAIN OVERVIEW

Blockchain is a decentralized system. It refers to the collective maintenance of a technical solution that maintains a continuous record file as a reliable database through decentralization. It was initially used extensively on Bitcoin [2]. II-A(Fig. 1) The block generation method of Blockchain is to collect and verify the data and then generate a new block through. We first describe the Blockchain consensus mechanism using Bitcoin as an example, its Blockchain consensus mechanism is a proof of work algorithm (POW). Each node competes based on their respective computing power to solve a SHA256 math problem that is complicated to solve but easy to verify. The first node that solves this problem will get the new block accounting right.

Blockchain data is stored on each node, then the nodes exchange information with each other over the network. Each node maintains an entire Blockchain data. The node will verify the received transactions and include them in the new

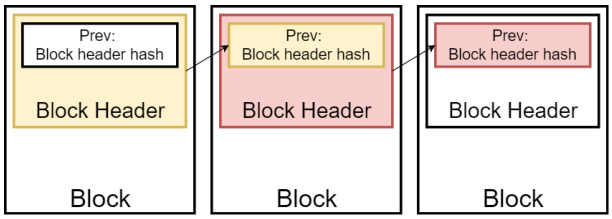

FIGURE 1. Connections between blocks in Blockchain.

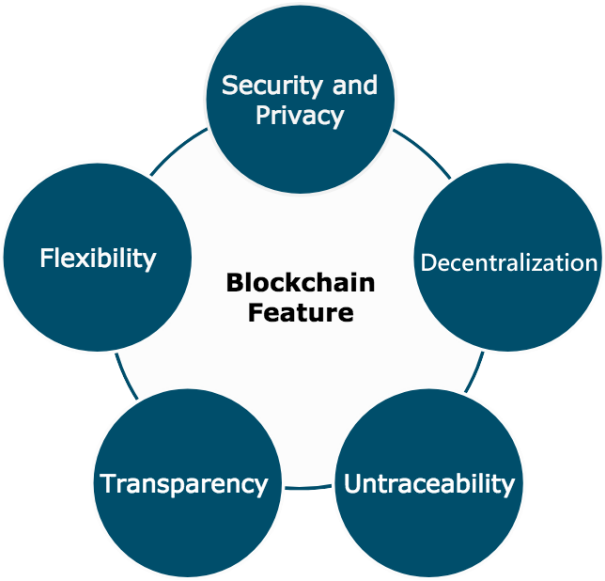

FIGURE 2. Characteristics of Blockchain.

block based on its own Blockchain data, and try to obtain the accounting rights of the new block in the above manner.

B. BLOCKCHAIN FEATURES

In today's social system, a large part of the economic behavior of individuals depends on trust where regularly two sides interact with a third party, thus forming a trust relationship. Usually, there is a mutual non-trust between these two parties that have long been based on trust guarantees provided by third parties, therefore it is important to take notice of the characteristics of Blockchain technology that help subvert the foundation of human transactions that have been conducted for thousands of years.

Using Blockchain one can create a data record system that does not depend on a trusted third party as a transaction intermediary, and that is openly shared and reliable at the same time. The characteristics of Blockchain technology are described in detail below. II-B(Fig. 2)

1) Security and Privacy: Public key encryption in cryptography is used in Blockchain to protect data security. Users can generate their own key pairs, including a private key and a public key. Private key is used to sign data, and public
key is used to verify the authenticity of the signed data. As long as the user prevents the private key from leaking, the data will remain secure.

In Blockchain, each user is anonymous, and each user can have multiple addresses. When the system is operating, only one address is used as the identification, and the anonymous address can hardly be mapped to the real person, thereby protecting the user's privacy.

2) Decentralization: Through decentralized operations and storage, each node of the Blockchain implements the verification, delivery, and management of information at the local side. Blockchain technology does not rely on an additional third-party control, has no centralized control, and is self-contained.
3) Untraceability: After a block has been determined in the Blockchain, it cannot be tampered with. Due to the following circumstances, once a block in the Blockchain is altered, it will be immediately detected and rejected by other nodes.
4) Transparency: The data in Blockchain is completely public and anyone can inquire. Within the information flow, one can clearly see who is passing data to whom as Blockchain maintains a continuous transaction log file.
5) Flexibility: The technology of Blockchain is open source and anyone can use it to modify it into their own version. There are already numerous flexible Blockchain platforms available, and users can also redevelop a new Blockchain platform if they desire so. Blockchain is an unlimited technology meaning that users can create multiple applications based on Blockchain.

C. BLOCKCHAIN 1.0 - BITCOIN

Bitcoin is a decentralized virtual currency that does not rely on specific currency institutions to circulate. By using the Blockchain consensus mechanism to trade virtual currency transactions, the problem of virtual currency security can be perfectly solved, such as the double-spending problem.

III-C(Fig. 3) Only when the user's private key is leaked, or forgotten, the user's Bitcoin will be lost.

Bitcoin is the first practical example of a Blockchain application. It has the following four characteristics:

- Decentralized peer-to-peer network
- Public transaction ledger
- Fixed currency circulation
- Decentralized transaction verification

Bitcoin is a representative of Blockchain 1.0. The information stored in the block is transaction data. It is used primarily as a decentralized electronic currency. Later, there were other research and development based on Bitcoin, such as color coin [3]. Also, there are some other almost same protocol electronic currency such as Litecoin [4].

D. BLOCKCHAIN 2.0 AND LATER VERSIONS - ETHEREUM

At the end of 2013, Vitalik Buterin published white paper of Ethereum [5], and yellow paper was published in 2014[6]. In July 2015 launched Ethereum frontier system and continued to improve it to this day. Ethereum is a Blockchain platform. Unlike the Blockchain technology used by Bitcoin, Ethereum is no longer limited to transaction records and is more effective and robust than its counterpart Bitcoin.

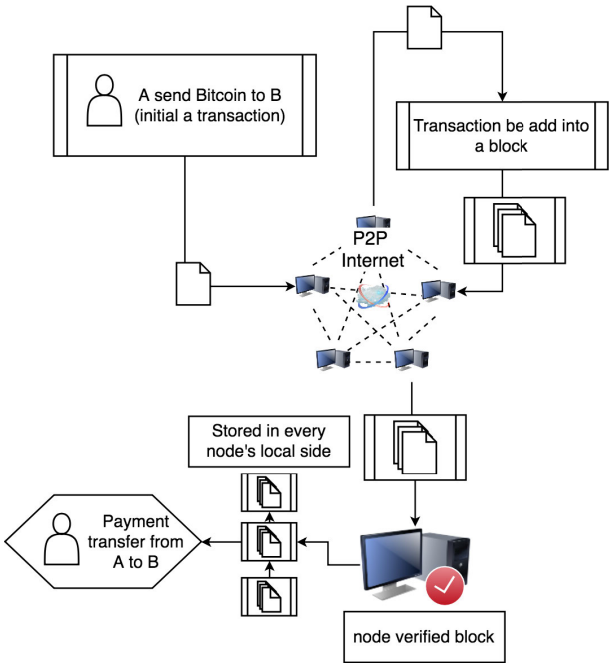

FIGURE 3. Flowchart of sending Bitcoin.

Ethereum is a Blockchain platform that can build smart contracts using a turing-completeness programming language. Anyone can write smart contracts or other decentralized applications on Ethereum. Users can set access permissions, transaction formats, state conversion equations, and so on, and build any desired rules.

Users of Ethereum will first write a smart contract using Solidity, then they will change their smart contract Solidity code into Ethereum bytecode, and add the bytecode into a transaction and deploy the transaction into the network. When miners of Ethereum receive the transaction, they will record the transaction in a block and run the bytecode in the Ethereum virtual machine each time a transaction of this smart contract is called. II-D(Fig. 4) To interact with a smart contract on Ethereum, the user has to send the information packaged in a transaction to communicate with the smart contract and interact with the smart contract by following the rule established within the smart contract. If successful, the smart contract will then have state changed on each miner's local storage. II-D(Fig. 5)

Ethereum is the representative of Blockchain 2.0. There also exist other Blockchain 2.0 and after applications such as Hyperledger [7]. In Blockchain 2.0 and after, there is no longer just transaction data stored in a block. It can be any information, flexibility is much higher than that of Blockchain 1.0.

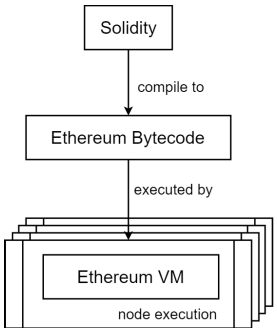

FIGURE 4. State change in Ethereum smart contract.

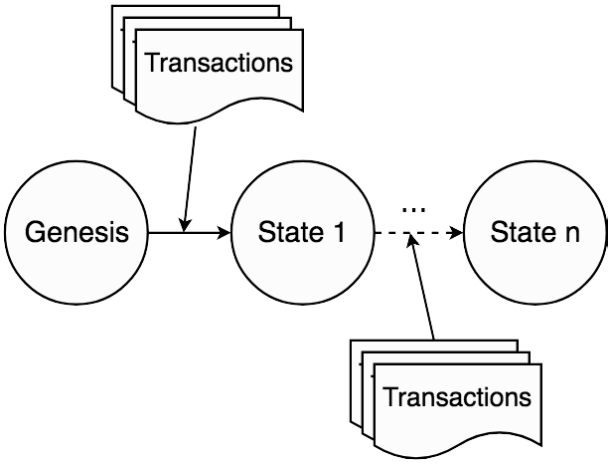

FIGURE 5. Transactions in Blockchain.

TABLE 1. Comparison between blockchain 1.0 and blockchain 2.0 and later versions.

Features	Blockchain 1.0	Blockchain 2.0 and after
Turing Completeness	No	Yes
State	Only two state	Multi-state
Block Time	Long	Short
Block Storage	Fixed script	Diversity of context height

E. COMPARISON BETWEEN BLOCKCHAIN 1.0 AND BLOCKCHAIN 2.0 AND LATER VERSIONS

Blockchain 1.0 and Blockchain 2.0 both use the Blockchain concept as a basis for their composition, but there are extensive differences in terms of implementation and usage. Table 1 lists the most crucial differences between Blockchain 1.0 and Blockchain 2.0 and later versions, clarifying why Blockchain 2.0 and later versions are used as the infrastructure for thesis.

- Turing Completeness: The script in Bitcoin is designed to not support loops so infinite loops can be prevented. On the other hand, Ethereum provides more flexibility in script writing, and Turing completenes as it employs a different method to prevent infinite loops.
- State: In Bitcoin, the unspent transaction output (UTXO) can have two states, spent or unspent and there is no opportunity for multi-stage contracts or scripts, which can keep any other internal states beyond that. Ethereum provides more flexibility to create such contracts by using externally owned accounts and contract accounts. In Ethereum, the multi-state can be defined by the user.
- Block Time: The creation of the blocks in Ethereum Blockchain takes 12 seconds, which is significantly faster than in Bitcoin Blockchain which takes nearly 10 minutes. The lesser the block time, the faster the transaction is established. Also, the security problems emerging from the shorter block time have already been solved by Blockchain 2.0 and later versions.
- Block Storage: In Bitcoin, only fixed scripts and fixed data can be recorded in a Block. On the other hand, self-implemented scripts can be stored in Ethereum's Blockchain, which means that thousands of different applications can be implemented on Ethereum.

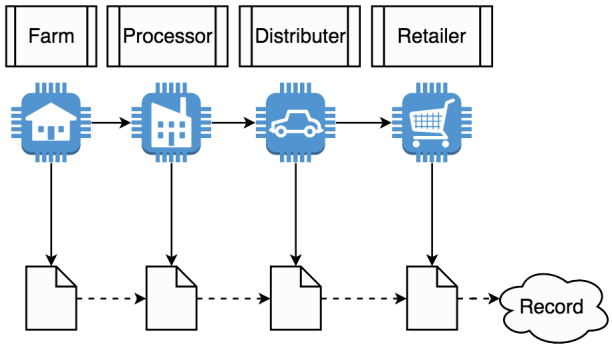

FIGURE 6. Supply chain data record on Blockchain.

We chose Ethereum which is a Blockchain 2.0 as our back-end platform. Due to the fact that if we needed to customize our smart contracts system, Ethereum will be a significantly better choice than Bitcoin as our system needs multi-state and flexibility in block storage to record data. The block time of our system also needs to be short so that the response time of trading can be acceptable by the customer standards. Additionally, by applying a Turing complete system writing a smart contract in the Blockchain is more straightforward as compared to previus versions.

III. RELATED WORK

There exist numerous Blockchain-based applications are gradually being developed. Some of the applications focus on payment verification, such as digital currency [8], stock trading [9], or financial securities. Some focus on combining Blockchain with the Internet of Things (IoT), such as recording the device data of IoT [10]. Other Blockchain decentralized applications such as game [11], gambling [12], on-line voting [13], car renting [14], and so on.

Here we will focus on a supply chain application, the recording data process on the supply chain is similar to our system's sale information recording. III(Fig. 6) The preceding research in supply chain management based on Blockchain is discussed as follows.

In [15] the authors provide the design principles for supply chain management on Blockchain. The authors indicated that counterfeit products are an important issue that modern brands with multinational supply chain networks always need

to acknowledge. By applying Blockchain in the supply chain data record, they can strictly monitor the flow of products.

Author of [16] analyses the advantages and disadvantages of binding RFID [17] and Blockchain technology on the supply chain, and cover the process of information management in Blockchain application. The author claims that given the Blockchain characteristics, the information recorded on the Blockchain can be completely trusted.

In the case study for product traceability [18], and the implemented system is named originChain. This system applies traceability of product by replacing normal centralized database with Blockchain data storage. The main idea of this system is to record the lab's product sample-testing results.

A product ownership management system [19] published in 2017 presents a system that implements Ethereum to provide the holding certificate of the consumer and combined the RFID of products to make sure that the product has its own identity stored in the Blockchain. However, the proposed scheme can not guarantee that the product the consumer purchased from the seller is not a counterfeited one. Hence, the product counterfeit problem is still unsolved.

On the business side, a company called Seal Network [20] is combining Blockchain technology and Near-field communication (NFC) to develop a product authentication platform. This company inserts NFC chips into each item and use them as the certificates of the product. The NFC data is uploaded into the company's Blockchain. However, using NFC chips is not suitable for all types of products. For instance, fresh food or small commodities. Furthermore, in this kind of system, customers still get the products from the sellers and not directly from the manufacturer and reasonably the consumers may have concerns trusting the sellers.

IV. SYSTEM DESIGN

We propose an irreproducible and complete product anti-counterfeiting system, which is based on Blockchain. In our scheme, manufacturers can use this system to store relevant information on product sales in Blockchain which is accessible to anyone. The total amount of sales that can be sold by the seller and the number of products currently left by the seller are transparent to users. The user can use the functions provided by our system to i.mmediately perform vendor-side verification, and this verification cannot be made. IV(Fig. 7)

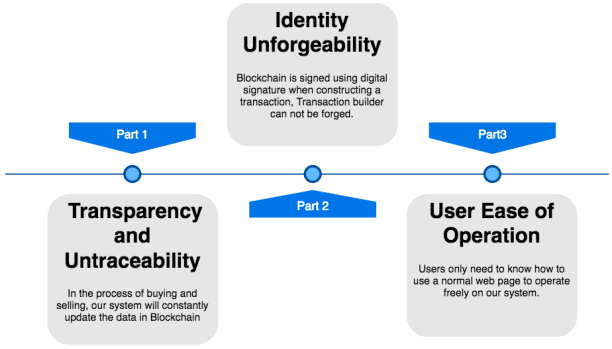

FIGURE 7. Design goal of our system.

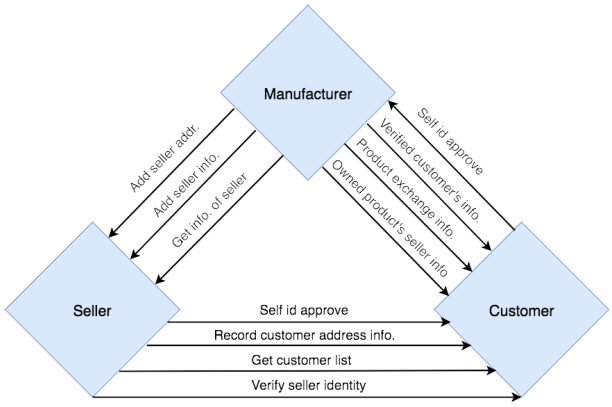

FIGURE 8. Basic model of our system.

A. BASIC MODEL

Our product anti-counterfeiting system based on Blockchain is composed of three roles, The Manufacturer Role, The Seller Role, and The Consumer role, we discuss them as follows: IV-A(Fig. 8)

1) Manufacturer Role: For the seller's part, the provided functions include adding new seller's address on contracts, adding the number of products that the seller can sell, and retrieving information on sellers so that the latest sales status can be retrieved. In the consumer's part, it is possible to inquire about the product the seller marketed to the consumer, and verify whether the product has yet been exchanged or confirm if the current status of the product has yet been verified by the consumer's public key certificate.
2) Seller Role: For the consumer's part, the seller can use the system's functions to encrypt the verification information with a private key, and the consumer can use the seller's public key to verify if the seller is what he claims to be. After buying and selling, the seller specifies the purchaser's address in the contract for the manufacturer to obtain the information. The seller can access information about his products, such as sales lists, and the quantity of his remaining stock.
3) Consumer Role: In the seller's part, the consumer can verify whether the seller has a sales relationship with the manufacturer and also verify whether the seller's stock hasn't been yet sold out. In the manufacturer's part, the consumers can prove that their identity is consistent with their address and in the case of a well-preserved contract address, the consumers can obtain individual purchase records and product status in their product.

B. OPERATION FLOW

In our design, the manufacturer is responsible for pushing seller information to the contract, including the number of products the seller can sell and the seller address. After the

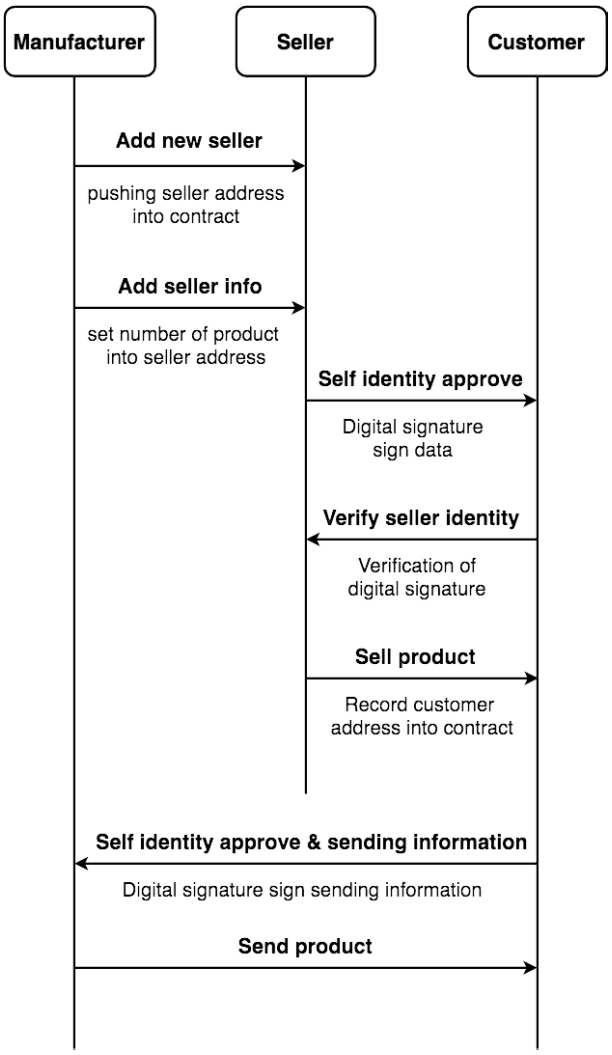

FIGURE 9. Flow chart of buying process in our system.

seller obtains the manufacturer's authorization, he can obtain a certain amount of recording rights for the products that he can sell on the contract. When the consumer purchases the product, the seller stores the consumer's address via the system to the contract to complete the transaction. Consumers can use the system to directly search for whether the seller is in the contract and whether there are unsold products available for trading. After purchasing, the consumer will provide the manufacturer with the information that the product wants to mail and encrypt it with the consumer's own private key. The manufacturer receives the encrypted data and will use the consumer's public key to restore it. If the information is consistent with consumer information, the manufacturer will send the product to the consumer and complete the shopping process. IV-B(Fig. 9)

V. IMPLEMENTATION

In this chapter, we explain the details of the design of our system, including a full description of the function and the user interface of the system. Our goal is to use the

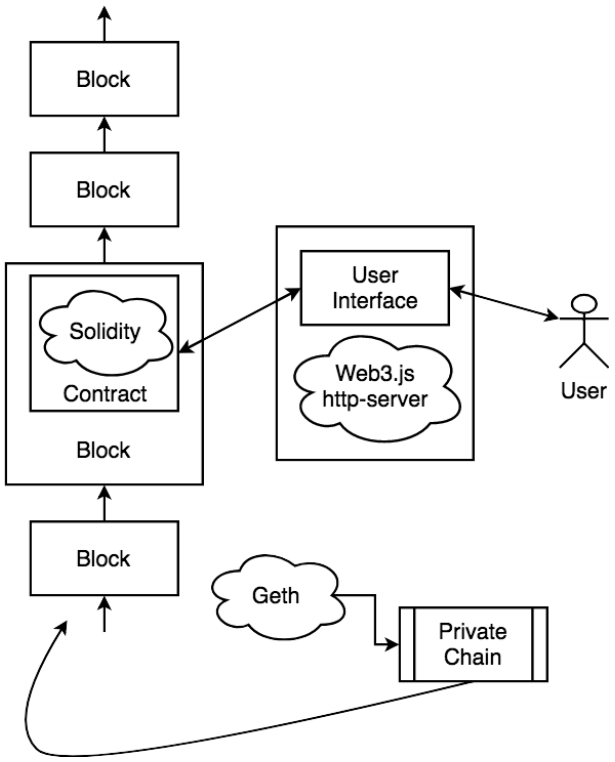

FIGURE 10. Architecture overview of our system.

Blockchain features to provide a more complete, convenient, and low-cost product anti-counterfeiting solution for manufacturing, sellers, and consumers.

A. PROGRAMMING LANGUAGE AND SYSTEM STRUCTURE

The proposed system uses Ethereum as the back end Blockchain operating system and uses Ethereum's proprietary programming language Solidity [21] as the high-level programming language for writing smart contracts. Solidity supports inheritance, libraries importing, etc. Solidity is designed for Ethereum Virtual Machine(EVM). Unlike Bitcoin's scripts, Solidity provides loops and it is Turing complete.

On the system, the public smart contract is based on Ethereum's Blockchain. In this paper, for ease of testing, we use Geth [22] to build a Private chain and push the smart contract on this Private chain, so that the Private chain simulates the situation of the Public chain. Plus use Mist [23] for account balance and contract information management.

The user interface seen by the user is a web page. The server side of the web page is made using the http-server [24] suite, which was provided by node.js [25] and web3.js [26] is used as the link between the smart contract and the user interface. The Private Chain and Address information can be connected after setting the server. The overall system relationship is shown in the following diagram. V-A(Fig. 10)

B. SYSTEM OPERATION

1) Login Process: Before establishing a connection to the system, the user has to choose which account to log in.

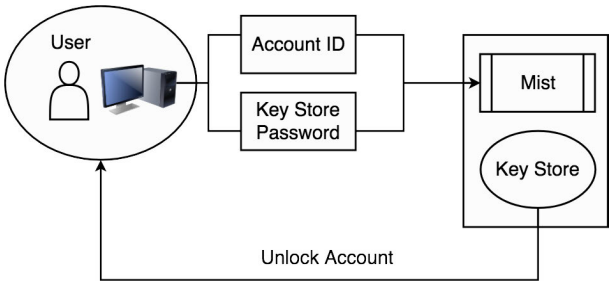

FIGURE 11. Log in process of our system.

TABLE 2. The data storage format in the system.

Struct Name	Data Type	Variable Name
Product	address	buyerAddress
	bool	exchange
Seller	address	sellerAddress
	unit	sellerProductNum
	uint	currentProductNum
	Product	products

The user's accounts are connected to accounts in Geth, as long as Geth is initiated, the user can choose the account which is also linked to the serial number of account list in Geth. Next, the user is required to type in the Keystore file, which is an encrypted file that holds the private key. Lastly, the user can key in the contract address and click the save button to set the basic information. V-A(Fig. 11)

2) Public Information of Contract: With the goal of information disclosure, the information about sellers is completely public. Our system provides smart contract data search functions, which can return the seller list, consumer list, all seller information, and the remaining number of products of each seller. The data storage format in the smart contract is shown in Table 2.
3) Adding New Sellers and Products Number: In our system, manufacturers can control the seller's information, including adding new seller addresses and as well adding the number of products that can be sold from a specific seller. The program in the smart contract will first check if the function setter is the manufacturer. If correct, the program will construct a seller Struct and set the sellers maximum number of products that can be sold, this amount can also be modified afterward.

Algorithm 1 SETNEWSELLER

Input: newSellerAddr, productNum
1: **if** (*SetterisManufacturer*) **then**
2: *Seller.sellerAddress* ← *newSellerAddr*
3: *Seller.sellerProductNum* ← *productNum*
4: **else**
5: *alert permission incorrect*
6: **end if**=0

4) Providing Exchange for Specific User Products: As customers provide proof of identity and the address to which the product is to be sent. The manufacturer will initially verify whether the identity is correct and then determines if the consumer product information is in the smart contract then will proceed to change the status of the product into exchange status. As a remark, this function will check whether the setter is the manufacturer. If not, the function will return without setting the value.

Algorithm 2 CHANGE

Input: buyerAddr, sellerAddr
1: *mapNum* ← *thesellerserialnumberincontract*
2: *productMgpNum* ← *unexchangeProductNumberIn SellerAddr*
3: **if** (*Setter is Manufacturer productMgpNum* > 0) **then**
4: *exchange* ← *true*
5: **else**
6: *alert error*
7: **end if**=0

5) Recording the consumer on Smart Contracts: When the transaction between seller and consumer is established, the seller will add the consumer address in the smart contract. Each seller has a product structure in the seller structure, the seller will put the consumer addresses into the product owner field. Additionally, the access rights of the seller's product owner field can only be set by the seller.
6) Identity Verification: Identity verification is one of the most important components in our system. Users in our system can use their address as their own representation. The address is defined within the last 20 bits of the user's public key. Whenever a user wants to make a change of current Ethereum contract state, the user has to sign the transaction with his private key to make a digital signature. As long as the user's private key is safe, there will be no other means to modify the user's identity.
To prove a user's identity to each other when necessary, our system allows users to use their private key to sign data and also provides the user with a function on which they can verify one another.

Algorithm 3 SETPRODUCTBUYER

Input: buyerAddr
1: *availiable* ← *theSellerHasAvailiableProductNumber*
2: **if** (*availiable* > 0) **then**
3: *Product.buyerAddress* ← *buyerAddr*
4: *Product.exchange* ← *false*
5: *Seller.currentProductNum* ← *Seller.currentProductNum* + 1
6: **else**
7: *alert error*
8: **end if**=0

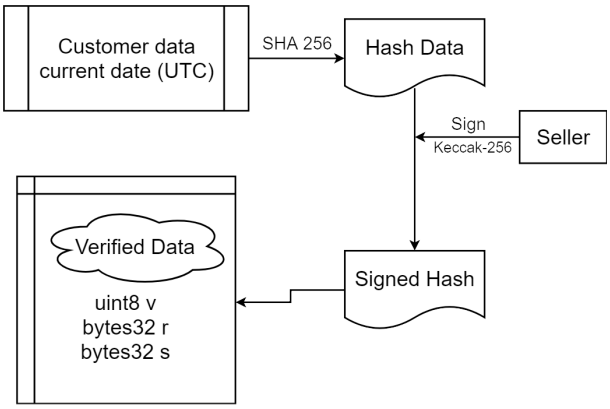

FIGURE 12. Flowchart of seller sign data.

Algorithm 4 SELLERSIGNDATA

1: *data* ← *concatenateTheInfoTheSellerWillSign*
2: *dataHash* ← *dataShaHash*
3: *signedHash* ← *sellerPrivateKeySignDataHash*
4: *separate sginedHash into v, r, s I > v r, and s are the values for the transaction's signature*

Before conducting a transaction with the seller, the consumer needs to ask the seller for a proof of identity. The consumer will provide the seller with a message that is to be encrypted. Then, the seller will call a function to encrypt the message, the function will concatenate the message and the current time, and will proceed to encrypt them. The system will then return the v, r, s, and the encryption time back. The seller will then send v, r, s, the encryption time, and the seller address to the customer. V-B(Fig. 12)

After the consumer acquires the essential information to verify the identity of the seller, the consumer will then call our system verification function to verify whether the seller identity is correct.

The customer will have to input the v, r, s, the encryption time, the seller address, and the message that the customer asked the seller to encrypt. The function will then return True if the seller is who he claimed to be. In contrast, return False if the seller is not legit. V-B(Fig. 13)

After being added into contract product owner field by seller, the consumer can send the shipment address information which was encrypted by the consumer's private key and the hash value of the encrypted information to be verified by the manufacturer. The customer has to provide the seller address, the recipient name of the product, the mail address of the recipient, and the phone number of the recipient. The system will sign these data with the consumer's private key, and provide these data to the manufacturer. V-B(Fig. 14)

The manufacturer will use our system verification function to verify whether the encrypted information is from the customer. The manufacturer will have to input the consumer address, the seller who sells a product to the seller, the recipient of the product, the mail address of the recipient, phone number of the recipient, and the encrypted verification data v, r, and s. The verification function will then return whether the customer is who he claimed to be. V-B(Fig. 15)

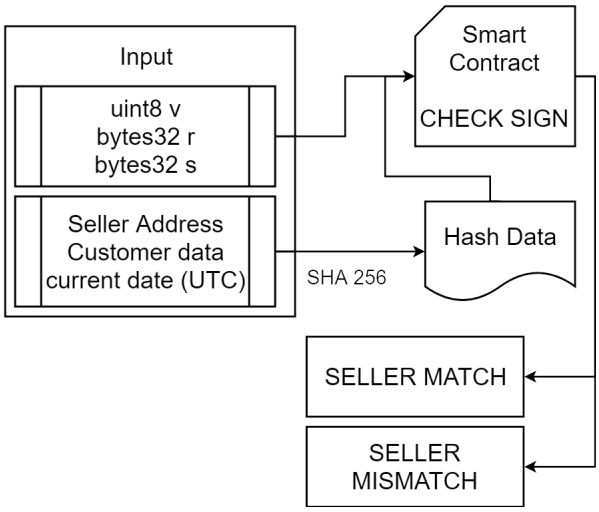

FIGURE 13. Flowchart of customer verified seller identity.

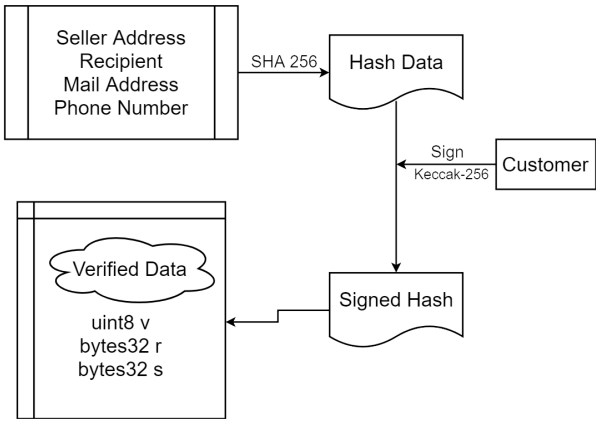

FIGURE 14. Flowchart of customer sign data.

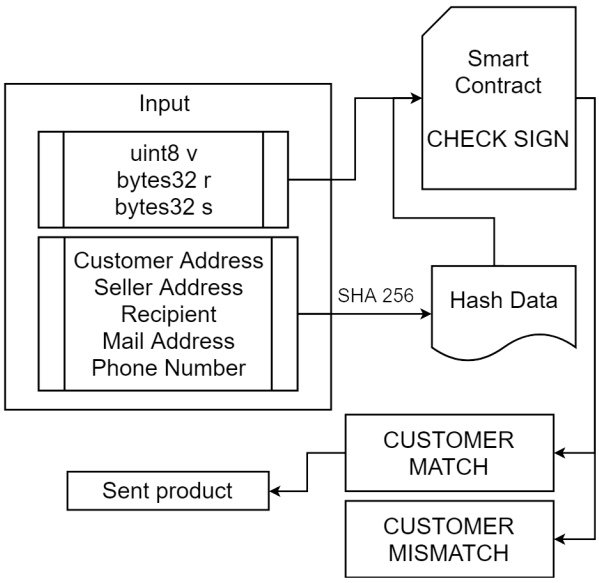

FIGURE 15. Flowchart of manufacturer verify customer's address.

Algorithm 5 SIGN

Input: v, r, s, sellerAddress, verificationData

1: *data* ← *concatenateTheInfoTheSellerSigned*
2: *dataHash* ← *dataShaHash*
3: *call Verification function in smart contract with parameter v, r, s I > v, r, and s are the values for the transaction′s signature*
4: **if** (*signaturematchtheselleraddress*) **then**
5: *The seller is verified*
6: **else**
7: *The seller is faked*
8: **end if**=0

Algorithm 6 SIGNDATA

1: *data* ← *concatenateTheInfoTheConsumerProvide*
2: *dataHash* ← *dataShaHash*
3: *signedHash* ← *consumerPrivateKeySignDataHash*
4: *separate sginedHash into v, r, s I > v r, and s are the values for the transaction′s signature*

Algorithm 7 CHECKSIGN

Input: v, r, s, sellerAddress, verif icateData

1: *data* ← *concatenateTheInfoTheConsumerSigned*
2: *dataHash* ← *dataShaHash*
3: *call Verification function with parameter v, r, s*
4: **if** (*signaturematchtheconsumeraddress*) **then**
5: *The consumer is verified*
6: **else**
7: *The consumer is not matched*
8: **end if**=0

VI. EXPERIMENT

In this section, we calculate the price to run our system's functionality in real terms, only with some state changes, execution movement or storage space use if the user decides to use Ether to pay to the miner. Consumer functions are entirely costless function, meaning that there is no contract data overwriting or storage memory expanding. The paying function will be paid by the manufacturer role or the seller role.

A. GAS

Gas is the pricing value of the execution work in Ethereum. When the user wants to make some state change in the smart contract, the user has to pay the corresponding state change gas. Different program operation cost different amounts of gas to run the operation. The operation cost of gas is a fixed number. For instance, each SHA3 operation costs 30 gas, and the paid for each byte in a LOG operation's data cost is 6 gas. There is a function cost used in our system exhibited as follows: (Table 3)

TABLE 3. Some of the fee in execution operations used in our system.

Operation description	Fee cost
Memory expanding for each addition word	3 gas
The contract creation operation	32000 gas
For every transaction	21000 gas
Each SHA3 operation	30 gas
Get transaction caller address	2 gas
Every zero byte of data for a transaction	4 gas
Contract suicide operation	5000 gas

TABLE 4. The relation between transaction accept speed and gas price in last 1500 blocks.

Transaction speed	Gas price
Lower than 30 minutes	3 gwei
Lower than 5 minutes	4 gwei
Lower than 2 minutes	14 gwei

TABLE 5. The relation between transaction accept speed and gas price in last 1500 blocks.

Statistic Set	Exchange rate
Earliest data in UTC time	498.02 US dollars
Highest data in the day	501.91 US dollars
Lowest data in the day	459.00 US dollars
Latest data in UTC time	477.49 US dollars

B. GAS PRICE

Gas price is the current price of gas. The gas price can be set by the user for any Wei. However, the higher the gas price is, the faster the transaction will be conducted in a block since the miner wants to earn higher rewards from the transaction and will give higher priority to the transaction with the higher price. EthGasStation [27] is an open-source project that aims to increase the transparency of gas price. This site estimates over last 1,500 blocks and finds the recommended gas price for the user. The following are the estimate data of gas price in last 1500 blocks from block 5785125. (Table 4)

The transaction is conducted in a block within five minutes. The statistics we choose are from EthGasStation. The recommended gas price it provides is based on the current network conditions and will have wave motion at different times. We choose the data with the latest update time in Block 5785125. (Table 4)

C. ETHER VALUE

As using Ether to pay for the procedure fee in the smart contract, we need to convert Ether into US dollars. Here we choose the CoinM arketCap [28] site to obtain the current conversion price.

(Table 5). The highest exchange rate is 501.91 US dollars per Ether, and the lowest exchange rate is 459.00 US dollars.

TABLE 6. The gas cost in executing system functions.

Function description	Transaction cost	Execution cost	Total cost
Deploy the contract of our system	1,548,966 gas	1,133,614 gas	2,683,580 gas
Add a new seller with an initial product number	89,417 gas	66,545 gas	155,962 gas
Add already exist seller product number	29,677 gas	6,805 gas	36,482 gas
Set a new product owner to consumer	56,509 gas	33,829 gas	90,338 gas
Consumer Check the seller's identity	34,624 gas (Cost only applies when called by a contract)	5,224 gas (Cost only applies when called by a contract)	39,848 gas
Manufacturer set product exchange for consumer	34,624 gas (Cost only applies when called by a contract)	5,224 gas (Cost only applies when called by a contract)	39,848 gas

We take the median of the highest and the lowest, which is 480.455 US dollars.

D. COST RESULT

The price, in reality, is the gas to run the corresponding function multiply by the gas price multiply by the current Ether value. Here we employ Remix [29] to calculate the gas need in our system function execution. Remix is a web browser IDE for developers in developing solidity Decentralized Application (DApp). Our system function execution gas spend is in (Table 6). The contract deploy function will only be called once for a new product. In our previous section prediction, the cost of contract deployment is 1.2893394289 US dollars. The other function cost will be 0.17415436749 US dollars, which means that each time a product sales process complete, it will cost 0.17415436749 US dollars. The overall cost of using our system for preventing counterfeited products is remarkably low cost to implement, and very straightforward to apply.

VII. CONCLUSION

This paper is the first Blockchain system that proposes a fully functional anti-product forgery system. By paying a very low transaction fee, users of our system no longer need to be concerned about the possibility of acquiring a counterfeited product.

Manufacturers can use the system to store relevant information on product sales in Blockchain which is accessible to everyone. The total amount of sales that can be sold by the seller and the number of products currently left by the seller are transparent. The user can use the functions provided by our system to immediately perform vendor-side verification. The system provides identity verification by using digital signatures. There are no other means to decrypt the private key of the key owner unless the key owner accidentally leaks his key.

In our system analysis result, the cost of the initial product record contract will only cost 1.2893394289 US dollars, and the cost of each product sale process will cost 0.17415436749 US dollars. Both costs are much lower than cooperating with reliable large chain stores and established Direct Selling Stores. Our system can effectively lower the threshold of the anti-counterfeiting of branded goods and provide the companies with limited financial resources an as well an easier approach to provide consumers with the confidence that they will not purchase counterfeited goods.

VIII. FUTURE WORK

The total cost of running an application on the Ethereum public chain is directly related to the code simplicity of the distributed application. The future work of this system can be proof of code simplicity. The customer can trust that the distributed application because of the simplicity of code, and no redundancy code in it will have additional consumption.

REFERENCES

[1] J. Leng, P. Jiang, K. Xu, Q. Liu, J. L. Zhao, Y. Bian, and R. Shi, "Makerchain: A blockchain with chemical signature for self-organizing process in social manufacturing," *J. Cleaner Prod.*, vol. 234, pp. 767–778, Oct. 2019.

[2] N. Alzahrani and N. Bulusu, "Block-supply chain: A new anti-counterfeiting supply chain using NFC and blockchain," in *Proc. 1st Workshop Cryptocurrencies Blockchains Distrib. Syst. (CryBlock)*, 2018, pp. 30–35.

[3] M. Rosenfeld. (2012). [Online]. Available: https://bitcoil.co.il/BitcoinX.pdf

[4] (2018). *Litecoin*. [Online]. Available: https://litecoin.info/index.php/Main Page

[5] (2019). *Github*. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[6] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, p. 1–32, Apr. 2014.

[7] (2018). *Hyperledger*. [Online]. Available: https://www.hyperledger.org/

[8] (2018). *Abra*. [Online]. Available: https://www.abra.com/

[9] (2018). *Tzero*. [Online]. Available: https://www.tzero.com/

[10] S. Panikkar, S. Nair, P. Brody, and V. Pureswaran, "ADEPT: An IoT practitioner perspective," IBM Inst. Bus. Value, New York, NY, USA, White Paper, 2015, pp. 1–18.

[11] (2018). *Cryptokitties*. [Online]. Available: https://www.cryptokitties.co/

[12] (2018). *Augur*. [Online]. Available: https://www.augur.net/

[13] (2018). *Agora Voting*. [Online]. Available: https://www.agora.vote/

[14] (2018). *Hirego*. [Online]. Available: https://www.hirego.io/

[15] S. Matthew English and E. Nezhadian, "Application of bitcoin data-structures & design principles to supply chain management," 2017, *arXiv:1703.04206*. [Online]. Available: http://arxiv.org/abs/1703.04206

[16] F. Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *Proc. 13th Int. Conf. Service Syst. Service Manage. (ICSSSM)*, Jun. 2016, pp. 1–6.

[17] S. Shepard, *RFID: Radio Frequency Identification*. New York, NY, USA: McGraw-Hill, 2005.

[18] Q. Lu and X. Xu, "Adaptable blockchain-based systems: A case study for product traceability," *IEEE Softw.*, vol. 34, no. 6, pp. 21–27, Nov./Dec. 2017.

[19] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, "A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain," *IEEE Access*, vol. 5, pp. 17465–17477, 2017.
[20] (2018). *Seal Network*. [Online]. Available: https://seal.network/
[21] (2018). *Solidity*. [Online]. Available: http://solidity.readthedocs.io/en/v0.4.24/
[22] (2018). *Geth*. [Online]. Available: https://geth.ethereum.org/
[23] (2018). *Myetherwallet*. [Online]. Available: https://www.myetherwallet.com/
[24] (2018). *HTTP-Server*. [Online]. Available: https://www.npmjs.com/package/http-server
[25] (2018). *Node.js*. [Online]. Available: https://www.myetherwallet.com/
[26] (2018). *Web3.js*. [Online]. Available: https://github.com/ethereum/web3.js/
[27] (2018). *ETH Gas Station*. [Online]. Available: https://nodejs.org/en/
[28] (2018). *Coinmarketcap*. [Online]. Available: https://coinmarketcap.com/
[29] (2018). *Remix*. [Online]. Available: https://remix.ethereum.org/

JINHUA MA received the M.S. degree from the School of Mathematics and Informatics, Fujian Normal University, China, in 2016, where she is currently pursuing the Ph.D. degree with the School of Mathematics and Informatics. Her research interests include cryptography and information security.

SHIH-YA LIN received the M.S. degree from the Department of Computer Science, National Tsing Hua University, Taiwan, in 2018. Her research interests include blockchain and information security.

XIN CHEN is currently pursuing the M.S. degree with the School of Mathematics and Informatics, Fujian Normal University, China. Her research interests include cryptography and password security.

HUNG-MIN SUN received the Ph.D. degree in computer science and information engineering from National Chiao-Tung University, Hsinchu, in 1995. He was an Associate Professor with the Department of Information Management, Chaoyang University of Technology, from 1995 to 1999, the Department of Computer Science and Information Engineering, National Cheng-Kung University, from 2000 to 2002, and the Department of Computer Science, National Tsing Hua University, Hsinchu, from 2002 to 2008. He is currently a Full Professor with the Department of Computer Science, National Tsing Hua University. He has published more than 200 international journal articles and conference papers. His research interests include network security, cryptography, blockchain, and automatic trading. He was the Program Co-Chair of the 2001 National Information Security Conference and the Program Committee Members of many international conferences. He was the Honor Chair of the 2009 International Conference on Computer and Automation Engineering, the 2009 International Conference on Computer Research and Development, and the 2009 International Conference on Telecom Technology and Applications. He is the General Chair of the ACM AsiaCCS'2020. He serves as the Editor Member of many international journals. He won many best paper awards in academic journal and conferences, including the Annual Best Paper Award from the *Journal of Information Science and Engineering*, in 2003, the Best Paper Award in MobiSys09, NSC05, NISC06, NISC07, CISC09, and ICS2010, ICMS 2019, IEEE ICKII 2019. He won the Y. Z. Hsu Scientific Paper Award from the Far Eastern Y. Z. Hsu Science and Technology Memorial Foundation, in 2010 and the Outstanding Research Award from World Congress on Information Technology Applications and Services, in 2015. He won the Award of Outstanding Professor in Electrical Engineering from the Chinese Institute of Electrical Engineering, in 2014.

YEH-CHENG CHEN (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Computer Science, University of California at Davis, Davis, CA, USA. His research interests include radio-frequency identification (RFID), data mining, social networks, information systems, wireless network artificial intelligence, the IoT, and security.

HUAXIONG WANG received the Ph.D. degree in mathematics from the University of Haifa, Israel, in 1996, and the Ph.D. degree in computer science from the University of Wollongong, Australia, in 2001. He has been with Nanyang Technological University (NTU), Singapore, since 2006, where he also served as the Head of the Division of Mathematical Sciences, from 2013 to 2015. He is currently the Deputy Director of the Strategic Centre for Research in Privacy-Preserving Technologies and Systems (SCRIPTS), NTU. He is author/coauthor of one book, nine edited books, and over 200 articles in international journals and conferences, covering various areas in cryptography and information security. He has supervised 27 Ph.D. students, and has served on the editorial board of nine international journals and as a member/Chair of the Program Committee for more than 100 international conferences. He received the Inaugural Award of Best Research Contribution awarded by the Computer Science Association of Australasia, in 2004. He was awarded the Minjiang Scholar by Fujian Province, China, in 2013. He was the Invited Speaker of ASIACRYPT 2017. He will serve as the Program Co-Chair of Asiacrypt 2020 and 2021.