Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6623251, 10 pages
https://doi.org/10.1155/2021/6623251

Research Article

A Blockchain-Based Vehicle Condition Recording System for Second-Hand Vehicle Market

You-Ting Jiang and Hung-Min Sun

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

Correspondence should be addressed to Hung-Min Sun; hmsun@cs.nthu.edu.tw

Received 26 October 2020; Revised 20 January 2021; Accepted 13 May 2021; Published 29 May 2021

Academic Editor: Laurie Cuthbert

Due to the cheap price and decent performance of used cars, the second-hand vehicle market is quite active in Taiwan. However, the vehicle information in the market of second-hand vehicles is centralized at present. In other words, the vehicle information is still provided by the second-hand vehicle dealers, which are not transparent and have no tampering cost. As the result, the authenticity, accuracy, and fairness of these second-hand vehicle data are worth discussing. To solve the above problems, this paper proposed a trusted vehicle data source system based on blockchain technology. The system is built on the Ethereum platform with the private chain structure. With the help of our platform, a trusted third party (e.g., maintenance plant and government branch) can record vehicle information into Blockchain so the integrity of vehicle information can be maintained. Customers can easily query the relevant vehicle information through our system interface and avoid receiving fake vehicle information. This system is a Proof of Concept (PoC), and the feasibility is examined by evaluating transactions per second (TPS) of the proposed system.

1. Introduction

The second-hand vehicle market is quite active in Taiwan. The reason is that although the second-hand vehicles are relatively old, they still maintain good performance due to the technological advancement of the automobile industry. Coupled with their relatively inexpensive price, many people will still consider used vehicle when purchasing a car. According to statistics from the Ministry of Transportation and Communications, Taiwan, there are about 600,000 transactions of vehicles a year in Taiwan's second-hand vehicle market [1], which is 1.5 times more than the new vehicle market. However, in the second-hand vehicle market, the most common situation which customers encounter is that they do not know whether the information of vehicle condition provided by the second-hand vehicle dealer is consistent with the actual condition and result in buying vehicles with problems (e.g., odometer fraud and accident vehicle). Although there are some vehicle information providers like Carfax [2], the situation of data fraud and tampering is still very serious.

Blockchain technology is a novel way to manage data. Due to its characteristics, a group of people (nodes) can work together to store, manage, and protect the blockchain without anyone trusting anyone, thereby eliminating the possible adverse consequences of centralized data control by third party. Moreover, each node would replicate the whole blockchain data through peer-to-peer networks, and so the information can hardly be destroyed. Therefore, we believe that blockchain technology can be applied to record the condition of second-hand vehicles.

This paper uses the blockchain architecture provided by Ethereum [3] to record the information of vehicle condition. By the properties of transparency and tamper proof of blockchain, customers can use the system proposed in this paper to check the actual vehicle condition without being worried misinformed by second-hand vehicle dealers.

2. Background

2.1. Introduce to Blockchain. The first successful blockchain application is Bitcoin [4]. In Bitcoin, blockchain acts as a publicly verifiable open ledger, which has the following properties:

(1) Transparency: The data on the blockchain is public and can be read by anyone; so, the record and flow of each transaction can be easily queried

(2) Decentralization: Bitcoin does not depend on additional third party to control the transactions. Each node in the Bitcoin network realizes verification, transmission, and management of transactions

(3) Tamper proof: Each block will contain the hash value of the previous block. Therefore, if someone tries to modify the data in the block, it will be immediately detected and rejected by other nodes, and so the blockchain cannot be tampered with

2.2. Smart Contract and Decentralized Application. The evolution of blockchain mechanism begins with trading of cryptocurrency, then gradually expands to the application of smart contract [5], and then enters the field of decentralized application. Below, we will introduce the concept of smart contract and decentralized application which are used to construct our vehicle condition recording system.

(1) Smart contract: The smart contract featured in Ethereum is written by a programming language called Solidity [6]. Before deploying, Solidity codes need to compile into Ethereum bytecode, being added into a transaction and send it to the network. Afterward, miners of Ethereum will verify the transaction and record it in a block to complete the deployment of a smart contract. Users of Ethereum can call a function of the smart contract by sending a transaction. When miners receive the transaction, they will verify it and record it in a block and run the bytecode of the smart contract in the Ethereum virtual machine (EVM) to complete the contract call. (Figure 1)

(2) Decentralized Application (DApp): DApp is defined as an application running on a Peer-To-Peer (P2P) network. In this network, every computer has the same status and same data and can execute the code of DApp, which giving DApp, following advantages

  (i) DApp running in the network will not be interrupted as long as there is at least one computer running in the P2P network

  (ii) When DApp is deployed in the blockchain network, the data stored in the block cannot be tampered with, even those who wrote and deployed DApp

In this paper, our vehicle condition recording system belongs to nonfinancial DApp, because the system does not involve the trading of vehicles.

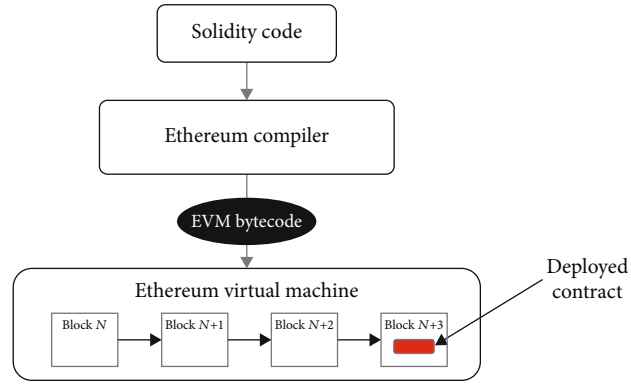

Figure 1: Ethereum smart contract execution flow.

2.3. Consensus Mechanism in Blockchain. Consensus is a mechanism to ensure that all the transactions occurring on the blockchain are genuine. It is often achieved by the contribution of other participants in the chain network. Below are three common types of consensus.

2.3.1. Proof of Work (PoW) [7]. PoW is the consensus mechanism used by Bitcoin and some other popular cryptocurrency networks. It requires participant nodes to solve a compute-intensive problem to receive the right to add new transactions to the blockchain. The participants who try to solve the problem are commonly called "miners" since they can get Bitcoins by verifying the blocks. However, this whole mining mechanism of Bitcoin needs high energy consumption and longer processing time.

2.3.2. Proof of Stake (PoS) [8]. PoS is another common consensus algorithm that serves as a low-energy consuming alternative to the PoW algorithm. Its algorithm is simple, and the node that has more cryptocurrency has more chance to verify transactions. The hypothesis behind this mechanism is that the node will have more motivation to maintain the genuineness of the system if they have more property on the chain. Since PoS does not include solving compute-intensive problems, it can significantly reduce the computing power than PoW. However, the scalability of transaction verification is still not improved in PoS.

2.3.3. Proof of Authority (PoA) [9]. PoA is a consensus algorithm proposed in 2017 by Ethereum cofounder and former CTO Gavin Wood. It introduces a practical solution for scale-up blockchain networks. Its concept is similar to PoS, but instead of using the number of cryptocurrencies, PoA decides the right of verifying transactions by nodes' reputation. In other words, transactions are verified by nodes that are selected as trustworthy entities. Since there are only a limited number of block validators, it makes PoA a highly scalable system. The PoA model enables companies to maintain their privacy while availing the benefits of blockchain technology. Microsoft Azure is an example where the PoA is being implemented.

2.4. Categories of Blockchain. At the present time, two types of blockchain are considered base on their publicity:

2.4.1. Public Chains. Public chains are open to anyone in the world and can be accessed, sent, received, and authenticated by anyone. It is a blockchain in which everyone can participate in a consensus process; thus, public chains are usually considered to be completely decentralized. However, the transaction on public chain is slow to update and iterate, and the initial set-up cost is very high if a developer wants to build his own chain with the current technology development framework [10]. If enterprises adopt public chain directly, they will be limited by the scaling problem, and the enterprise demand cannot be met.

2.4.2. Private Chains. A private chain is a completely private blockchain. Its write access is restricted to one organization. Access rights are either open to the public or restricted to a certain extent. The entire network is jointly maintained by the member organizations, and access to the network is generally through the member organizations' gateway nodes. With the consensus process controlled by preselected nodes, this type of blockchain is considered to be partially decentralized. However, at the expense of partial decentralization, private chains have characteristic of fast transaction, privacy-protecting, and having very low transaction costs and low set-up costs [11]. At the same time, it retains blockchain's tamper-evident nature.

3. Related Work

There are lots of DApps that have been developed since the emergence of blockchain [12]. Some of them deal with financial transaction issues, such as decentralized exchange [13]. Some deal with the issue of personal data ownership on the cloud storage platform [14, 15]. Other DApps include supply chain application [16, 17], insurance application [18], Internet of Things (IoT) with blockchain [19, 20], game application [21], and secondary radio spectrum trade [22]. Our system focuses on data recording, and this type of application is similar to the application of product biography or supply chain.

In [16], the authors propose a framework for blockchain-based supply chain quality management. This framework consists of blockchain, smart contracts, and various IoT sensors. Blockchain provides a safe distributed ledger with various quality information, assets information, logistic information, and transaction information. Smart contracts bring privacy protection, automation, and intelligence into this system, while IoT sensors gather various data from the real world. By using the framework, each piece of information in the production process of the product will be recorded on blockchain, including temperature, humidity, positioning, and other information in the process of product transportation. Record all information in the production process of the product in blockchain can help users quickly understand the real information of the product by means of the blockchain's tamper-proof characteristic. This way of creating a biography for a product can also be used in recording vehicle conditions.

Another research [23] is a big data-sharing system for the car industry in China based on blockchain. They have set up an automotive industry alliance in China and published a white paper, which mainly aims at establishing a blockchain platform for recording vehicle information. At the end of 2018, they officially released this blockchain platform called Engine Chain (EGCC), hoping to record the information of the whole life cycle of a car through this blockchain platform. This research is similar to what our system wants to do, but our system is slightly different from EGCC. First of all, to query car information or execute other operations on EGCC, users need to spend EGCC coin. On the other hand, our system would not charge for querying data. The second difference is the group that has record permission. EGCC allows all registered users to record data, but our system only allows trusted third party, which avoid malicious users trying to tamper data. The third difference is the consensus algorithm used. EGCC uses validated DPoS (VDPoS) [24], which is an algorithm proposed in the Insight Chain [24]. It is a modified PoS algorithm which can improve speed. Table 1 is a simple comparison between our system and EGCC.

Therefore, in this paper, we use blockchain technology to achieve the purpose that vehicle condition information cannot be tampered with, so as to help customers to have authentic information of vehicles for reference when choosing second-hand vehicles.

4. System Design

4.1. Design Goal. Our system is designed according to the following three main goals.

(1) Transparency of vehicle condition information: Our system is based on the PoA consensus of Ethereum so that the condition information of vehicles can be stored on the private blockchain of Ethereum by specific users, and the information recorded on Blockchain is transparent to everyone in this private network

(2) Access control of smart contract and tamper proof of blockchain information: Because our system must rely on the help of some specific users (e.g., government branch and maintenance plant) to record or update the information of vehicle condition to change the state of smart contract, we must only allow these specific users to update the state of smart contract; so, we use the conditional branch in a smart contract to judge whether the transaction sender has the permission to change the state or not, so as to achieve the access control of the smart contract

(3) User friendly of operation: We also design a web user interface (UI) for the system. Each user can use the web page to interact with the contract function to record or update the information of vehicle condition without having any background knowledge of smart contract. We design different web UI for users depending on each user's identity to record or read different information on the contract

4.2. System Model. There are many roles in the proposed system (Figure 2), and the duty of each different role will be explained in the following paragraphs.

Table 1: Comparison between the proposed system and EGCC.

	The proposed system	EGCC
Blockchain platform	Ethereum private chain	Engine Chain
Query fee	Free for query	Purchase for unlocking information
Record permission	Trusted third party	Registered user
Consensus algorithm	PoA	VDPoS

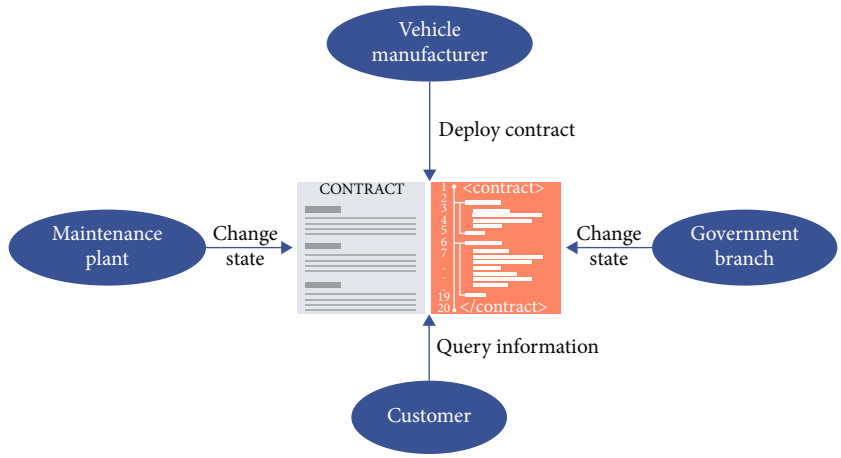

Figure 2: System model.

(1) Vehicle manufacturer: For vehicle manufacturers, the only function provided is the deployment of new contracts to the blockchain. After a manufacturer completes a car build, they can use the proposed system to deploy a smart contract that contains initial information about the vehicle. Initial information includes car model, year of release, mileage, and engine capacity.

(2) Maintenance plant: The maintenance plant can use the functions of the proposed system to record the maintenance history of vehicles. A maintenance record includes two types of information, one is the maintenance time, and another is the maintenance description. For instance, the maintenance plant can record the mileage of the vehicle to prevent customers from suffering from odometer fraud. In addition, before recording the information to blockchain, the owner ID of the vehicle is needed. Since the owner ID is a secret from the vehicle owner, this approach ensures that the maintenance plant can only record vehicle information with the vehicle owner's consent

(3) Government branch: The government branch can use the proposed system to record information. First, the Motor Vehicles Office, the government branch that is in charge of vehicles in Taiwan, can update the ownership history when the owner of a vehicle is changed. This information can make customers go through the ownership history of a vehicle and bring it into assessment. Motor Vehicles Office also has the permission to maintain the mapping table which maps the Vehicle Identification Number (VIN) to its Owner ID and only they can maintain this table. VIN is like an ID card number of a vehicle; thus, each vehicle has a different VIN. Second, when dealing with a traffic accident, the traffic police can record the time of the accident by our system, and this allows customers to understand the accident history of a vehicle and prevents them from buying an accident car. Third, when a vehicle is scrapped, the contract which represents the vehicle needs to be destructed by triggering the self-destruct function in the contract. Only Motor Vehicles Office has the permission to trigger the function

(4) Customer: Customers can use our system to query about the information of vehicles, including initial information, accident history, and maintenance records. Customers can simply enter the VIN of the vehicle and the token acquired from a second-hand vehicle dealer on the web page to query the information of the vehicle. Besides, when a general user wants to query the information of his vehicle, he can just input the VIN and owner ID to get his vehicle information rather than input a token. Customers do not need to have an account to query about the information of vehicles, as long as the customer inputs the correct token or owner ID, and he has the permission to read the information from blockchain. Moreover, they do not have the permission to change the state of smart contracts

(5) Second-hand vehicle dealer: The second-hand vehicle dealer in the proposed system do not have the permission to record or update the information of vehicles. They are only responsible for generating a

query token for the customer who wants to query the vehicle information. After inputting this token by the customer, our system will verify the token. If it is valid, then the system will return the vehicle information

(6) Authority node: As mentioned before, our system is based on the PoA consensus of Ethereum blockchain called Clique. Different from the PoW consensus, there has no miner in the PoA network. The accounting right of blocks is handled by predefined authority nodes. By using the PoA consensus Clique, transactions can be verified and accounted into the new block by the authority nodes without costing any fee

4.3. Life Cycle of Vehicle. In our design, the vehicle manufacturer is responsible for pushing the initial information of each vehicle to each contract, including car model, released year, mileage, and engine capacity. After the sale of the vehicle, the government branch (Motor Vehicles Office) needs to update the ownership history and change the owner ID of the vehicle on the contract. When using the vehicle, every using data that includes accident history, maintenance history, and mileage will be recorded or updated in the contract by the government branch and maintenance plant, respectively. When the owner of the vehicle sells the car to a second-hand vehicle dealer, Motor Vehicles Office also needs to update the ownership history. Customers can use our system to query the information of each vehicle by a valid token at any time in the life cycle of each vehicle until the vehicle is scrapped. When the vehicle is scrapped, the contract that represents the vehicle is also destructed by the self-destruct function of the contract (Figure 3).

5. System Implementation

In this chapter, we will focus on the details of the proposed system, including a full description of each function and the user interface of the system. In the first section, we explain the structure of the system and what software and tools are used. In the second section, we describe the functions of the proposed system in detail.

5.1. System Structure. The proposed system is performed on a private blockchain environment using Go-Ethereum (Geth) [25] client implementation. Clique, a PoA Geth implementation, is used as the back-end blockchain operating system, and Solidity is used as the programming language to write the smart contract. The system structure includes the following two parts:

(1) Private chain setup: Puppeth [26] is used to set up the configuration of the private chain. After the setup, Puppeth will generate a file that includes information of the configuration such as block time, gas limit, and authority node setup. Geth will later use this file to build up the private chain environment

In the configuration, we create different nodes for different identities of participants in the proposed system including vehicle manufacture, maintenance plant, and government branch for account management. In addition to the above nodes, three authority nodes are set up for system simulation.

(2) User interface (UI): The UI of the proposed system is a web page. The server side is constructed by the http-server suite provided by Node.js [27]. Web3.js [28] is used to link the front-end UI to the back-end smart contract. After the setup, private chain and account information can be connected

The system structure is shown in the following figure (Figure 4).

5.2. System Operation

(1) Log-in process: Since there are different roles of users in the proposed system, each user must choose which role to log in. Every user's account is connected to the accounts in Geth nodes. Different roles lead to connect to different Geth nodes. Because of the serial number of the account list in the Geth node, users can just input the serial number as the account number instead of inputting the account address (Figure 5)

(2) Operation of customer: Because of avoiding anyone that can query the vehicle information without any permission, the customer needs to input a token which is generated by second-hand vehicle dealer for token authentication. After passing the authentication, the customer can get the information of the vehicle. For the authentication purpose, we keep an array in smart contract to record these valid tokens for authentication. The customer needs to input the VIN of the vehicle and a token generated by a second-hand vehicle dealer to query its information. We use a smart contract to keep a mapping table to store each VIN and corresponding contract address; thus, the customers do not need to input complicated contract address string. Our system provides information query function in smart contract which can return car model, released year, engine capacity, mileage, ownership history, accident history, maintenance history, and so on

(3) Operation of vehicle nanufacturer: In the proposed system, the vehicle manufacturer is responsible for deploying a new contract that represents a vehicle. After logging in the system, the vehicle manufacturer can input the initial information of the vehicle and deploy the contract (Figure 6). After the deployment, the vehicle manufacturer will update the mapping table which stores each VIN and corresponding contract address. In the proposed system, only they can deploy a contract and update the mapping table

(4) Operation of maintenance plant: The maintenance plant in our system is responsible for recording maintenance history and mileage of vehicles. Before recording the information, our system will require that the maintenance plant should input the owner ID of the vehicle at first. If the owner ID is correct,

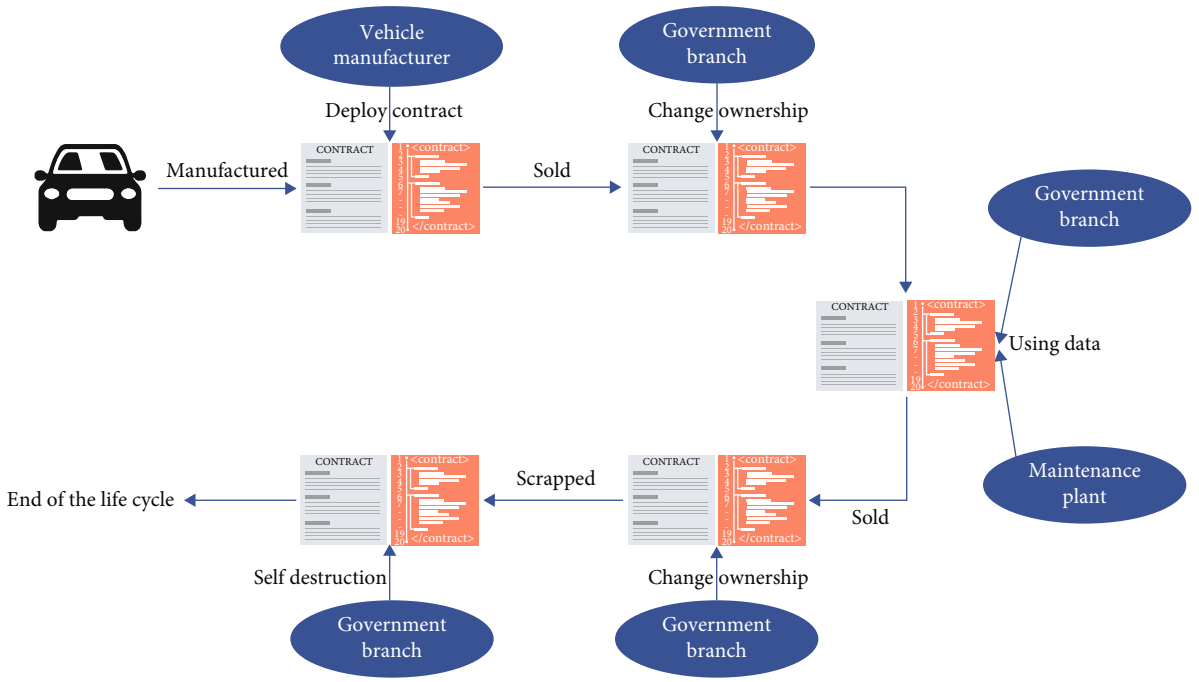

Figure 3: The life cycle of a vehicle in our system.

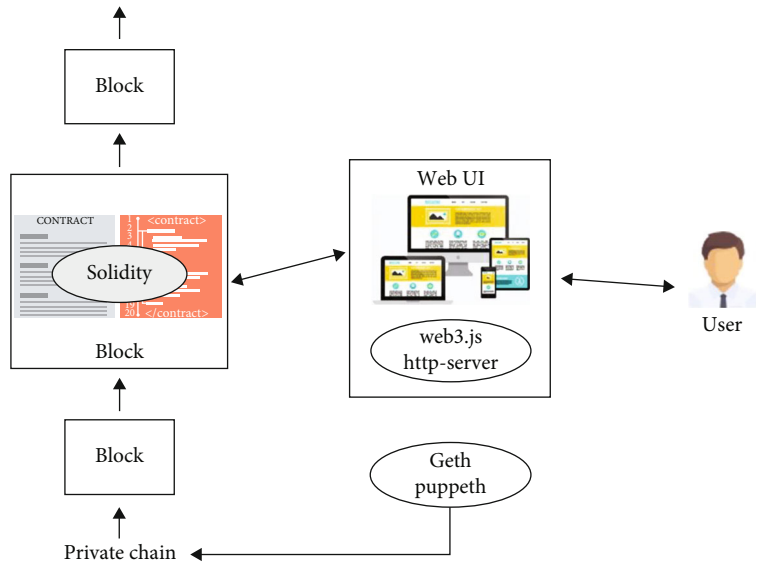

Figure 4: The overview of the proposed system structure.

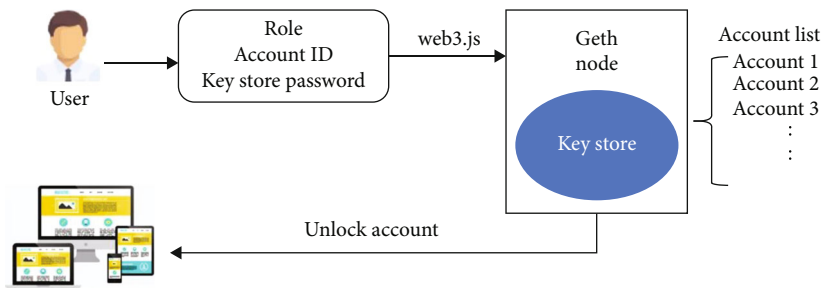

Figure 5: The log-in process of the proposed system.

```
var car = carContract.new(
        /* input of constructor */
        {
                from: web3.eth.accounts[accountNum],
                data: 'byte code of the contract',
                gas: '4700000'
        }, function (e, contract){
                console.log(e, contract);
                if (typeof contract.address !== 'undefined') {
                    alert("success! address: " + contract.address.toString);
                    document.getElementById("address").innerText = contract.address.toString();
                }
            }
        )
```

Figure 6: The code of deploying a contract.

then they can start recording the vehicle information. We use the mapping table *VIN-to-OwnerID* to help us for authentication.

When recording the information, the program in smart contract will check whether the role of the transaction sender is the maintenance plant that exists in the account list of the node. If the identity of the transaction sender is authentic, the program will append the time and the description of maintenance history to the Struct array *maintainHistories* and update the mileage.

(5) Operation of government branch: The operation of the government branch is similar to maintenance plant. For the ownership history, the program in smart contract which is triggered by Motor Vehicles Office will update the ownership number when the owner of the vehicle is changed. Motor Vehicles Office is also responsible for maintaining the mapping table *VIN-to-OwnerID*, and when the owner of a vehicle is changed, only Motor Vehicles Office can update the mapping table. We use SHA256 to store the hash value of owner ID in the mapping table in smart contract rather than store the plaintext of owner ID.

For the accident history, the program in smart contract which is triggered by police will append the time of the accident to the String array *accidentHistory*. Finally, when the vehicle is scrapped, Motor Vehicles Office needs to trigger the self-destruct function of the contract, and we use the instruction *selfdestruct()* in Solidity to destruct a smart contract. Before all of the above operation, the program in smart contract still first checks whether the role of the transaction sender has the permission to change the state of the contract or not.

(6) Operation of second-Hand vehicle dealer: The second-hand vehicle dealer does not have the permission to record or update the information of vehicles. They are only responsible for generating a query token for the customer who wants to query the vehicle information. For the token generation, our system concatenates the password of the second-hand vehicle dealer and current time as the input string and output the hash value (token). Then, the dealer will write this token into an array in the smart contract

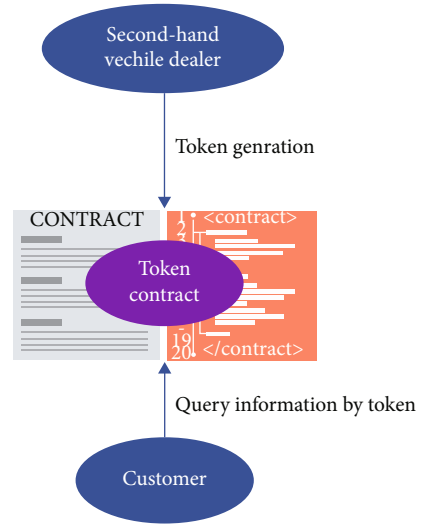

Figure 7: The operation of second-hand vehicle dealer.

Every token generated by a second-hand vehicle dealer has its lifetime. The proposed system will delete it by sending a transaction to delete the token in three minutes after its generation. This ensures that every token is timeliness and difficult to reuse (Figure 7).

6. System Evaluation

6.1. Gas Limit and Transaction Throughput. The analysis is performed on a PoA private Blockchain development based on the Ethereum platform, using Go-Ethereum (Geth) client implementation. Since there is no fee for sending transactions in a private Blockchain environment, we focus on the throughput of transactions. The proposed system is feasible only if the throughput of transactions is greater than the daily transactions.

Ethereum platform prevents transaction spamming and rewards block miners by charging a gas fee on transactions. There are three important concepts in the Ethereum gas mechanism, gas, gas price, and gas limit.

(1) Gas: Each bytecode instruction in the EVM execution process requires a corresponding amount of gas, and the more complex the logic, the more gas is required.

Table 2: The TPS of each function execution using block time = 5 seconds.

Function description	Transaction cost	TPS	Transactions per day
Deploy the contract	2,738,400	0.58	50,482
Record the accident time	63,522	25.18	2,176,253
Record the maintenance history	88,448	18.08	1,562,952
Update the mileage	27,897	57.35	4,955,371
Update the ownership history	41,814	38.26	3,306,069
Average (without deploy contract)	55,420	34.71	3,000,161

Table 3: The TPS of each function execution using block time = 10 seconds.

Function description	Transaction cost	TPS	Transactions per day
Deploy the contract	2,738,400	0.29	25,241
Record the accident time	63,522	12.59	1,088,126
Record the maintenance history	88,448	9.04	781,476
Update the mileage	27,897	28.67	2,477,685
Update the ownership history	41,814	19.13	1,653,034
Average (without deploy contract)	55,420	17.36	1,500,080

Table 4: The TPS of each function execution using block time = 10 seconds.

Function description	Transaction cost	TPS	Transactions per day
Deploy the contract	2,738,400	0.19	16,827
Record the accident time	63,522	8.39	725,417
Record the maintenance history	88,448	6.03	520,984
Update the mileage	27,897	19.12	1,651,790
Update the ownership history	41,814	12.75	1,102,023
Average (without deploy contract)	55,420	11.57	1,000,054

The table of gas spend to corresponding bytecode is defined in Ethereum yellow paper [3]

(2) Gas price: Gas price means how much you are willing to pay for each gas, in unit Gwei, which equals to 10^{-9} Ether, the basic unit of coin in Ethereum

(3) Gas limit: The gas limit is the maximum amount of gas you are willing to buy for the transaction. The reason for the gas limit is that no one knows how much gas will be used in the execution of a smart contract until it is actually executed, and so the gas limit is meant to cap the fee for each executed smart contract

Transactions per second (TPS) is a common criterion to evaluate how fast can a transaction be verified on the blockchain. The theoretical maximum TPS can be calculated using the following equation [29] (Eq. (1)), where *GasLimit* is the maximum number of units of gas user willing to spend on a transaction. *TxGas* is the gas needed to compute the simplest transaction, and *BlockTime* is the time required to create the next block in a chain.

$$\mathrm{TPS}_{\max} = \frac{\mathrm{GasLimit}}{Tx\mathrm{Gas} \times \mathrm{BlockTime}}. \quad (1)$$

6.2. Evaluations. The gas limit and the block time could be set as a configuration parameter. We set the gas limit to 8,000,000 gas by default configuration in the proposed system, which is the same as the gas limit of Ethereum main network [3].

We test the throughput using different block times, 5 s, 10s, and 15 s, respectively. For the gas cost of each transaction, we use Remix [30] to calculate the gas cost of function execution in the system. Remix is a web browser integrated development environment (IDE) for developers in developing Solidity smart contract and DApp. The TPS is calculated using Eq. (1). Transactions per day are then calculated by TPS multiplying seconds within a day. Tables 2–4 report the TPS of each function execution using different block times in the proposed system. For display conciseness, the TPS column is rounded to two decimal places, while the transactions per day column were computed using unrounded TPS.

According to Tables 2–4, it can be observed that the acceptable number of transactions per day of deploying contract is lower than the others since it needs higher gas cost. Besides the deploy contract function, the TPS of other functions are good enough. Therefore, we only focus on the TPS of deploying contract, since lower block times were found to be strongly related to the higher number of lost blocks per valid block [29], which leads to a bad performance of the Blockchain. For avoiding this problem, we set the block

time to 15 seconds by default configuration in our system. Assume that we have 600,000 vehicles (yearly second-hand vehicle transaction amounts in Taiwan) to deploy to the blockchain at the beginning, the required time of deploying these contracts is about 35 days, and it is acceptable.

7. Limit and Challenges

This paper's implementation relies on the existing platform (Ethereum). Thus, the functionality is restricted by the functions their system provides. In addition, this paper's method is a proof of concept; thus, it might still have problems when it runs online, such as how to choose a proper trusted third party to record the chain. Also, PoS is not a fully centralize consensus algorithm; thus, it is not as decentralized as Bitcoin which uses PoW. However, it still could avoid second-hand traders hold car information by themselves. So, it is still could solve this problem.

8. Conclusions

This paper builds a trusted vehicle data source system based on blockchain technology. With the help of trusted third parties to record vehicle information to our platform instantly, customers no longer need to worry about the problem of getting untrustworthy vehicle condition information from second-hand vehicle dealers. Moreover, because of the property of blockchain, we can ensure that the information of vehicle condition in the system is transparent to everyone and cannot be tampered with.

Vehicle manufacturers can use the system to deploy a smart contract that represents a vehicle. The maintenance plant and government branch can record or update the vehicle condition information by using the proposed system. Customers can easily query the relevant vehicle information through the system interface. All of the above operations can be done without costing any fee.

In the evaluation of the proposed system, the result shows that the transaction throughput is enough for daily transactions.

Data Availability

The code implementation used to support the findings of this study were supplied by Hung-Min Sun under license and so cannot be made freely available. Requests for access to these data should be made to Hung-Min Sun: hmsun@cs.nthu.edu.tw.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This study was funded by the Ministry of Science and Technology, Taiwan [grant numbers MOST 106-2221-E-007-026-MY3, MOST 107-2221-E-007-015-MY3, and MOST 108-2218-E-001-001].

References

[1] "Ministry of Transportation and Communications, Taiwan. Commonly Used Transportation Statistics," 2020, https://stat.motc.gov.tw/mocdb/stmain.jsp?sys=100&funid=emenu.

[2] "Carfax," 2020, https://www.carfax.com/.

[3] G. Wood, "Ethereum: a secure decentralised generalized transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[4] S. Nakamoto and A. Bitcoin, "A Peer-to-Peer Electronic Cash System," 2008.

[5] "Smart contract," 2020, https://ethereum.org/en/developers/docs/smart-contracts/.

[6] "Solidity," 2020, https://docs.soliditylang.org/en/v0.7.4/.

[7] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols(extended abstract)," in *Secure information networks*, pp. 258–272, Springer, Boston, MA, 1999.

[8] J. Kang, Z. Xiong, D. Niyato, P. Wang, D. Ye, and D. I. Kim, "Incentivizing consensus propagation in proof-of-stake based consortium blockchain networks," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 157–160, 2019.

[9] "Proof of authority," 2020, https://github.com/PoAnetwork/wiki/wiki/PoA-Network-Whitepaper/.

[10] D. Guegan, *Public Blockchain Versus Private Blockhain*, 2017.

[11] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, pp. 557–564, Boston, MA, USA, 2017.

[12] "State of the DApps," 2020, https://www.stateofthedapps.com/zh/stats/.

[13] "Idex - decentralized ethereum asset exchange," 2020, https://idex.market/eth/idex/.

[14] N. Kaaniche and M. Laurent, "A blockchain-based data usage auditing architecture with enhanced privacy and availability," in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, pp. 1–5, Cambridge, MA, USA, 2017.

[15] G. Zyskind and O. Nathan, "Decentralizing privacy: using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, pp. 180–184, San Jose, CA, USA, 2015.

[16] 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)*, pp. 172–176, Shanghai, China, 2017.

[17] A. Iftekhar, X. Cui, M. Hassan, and W. Afzal, "Application of blockchain and Internet of Things to ensure tamper-proof data availability for food safety," *Journal of Food Quality*, vol. 2020, Article ID 5385207, 14 pages, 2020.

[18] I. Nath, "Data exchange platform to fight insurance fraud on blockchain," in *IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 821–825, Barcelona, Spain, 2016.

[19] P. Urien, "Blockchain iot (biot): a new direction for solving internet of things security and trust issues," in *2018 3rd Cloudification of the Internet of Things (CIoT)*, pp. 1–4, Paris, France, 2018.

[20] Y. Wu, L. Song, L. Liu, J. Li, X. Li, and L. Zhou, "Consensus mechanism of IoT based on blockchain technology," *Shock and Vibration*, vol. 2020, Article ID 8846429, 9 pages, 2020.

[21] "Etheremon," 2020, https://www.etheremon.com/.

[22] M. A. Khan, M. M. Jamali, T. Maksymyuk, and J. Gazda, "A blockchain token-based trading model for secondary spectrum markets in future generation mobile networks," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 7975393, 12 pages, 2020.

[23] "Enginechain white paper," 2020, https://white-paper.oss-cn-beijing.aliyuncs.com/v.3.0.0/Engine%20V.3.0.0%20-%20%20English%20Version.pdf/.

[24] TEAM, "Insight Chain Founding," *Insight Chain Technical White Paper V1. 0*, 2019.

[25] "Geth," 2020, https://github.com/ethereum/go-ethereum/wiki/.

[26] "Puppeth," 2020, https://github.com/puppeth/.

[27] "Node.js," 2020, https://nodejs.org/en/.

[28] "Web3.js," 2020, https://web3js.readthedocs.io/en/1.0/.

[29] "On ethereum performance evaluation using PoA," 2020, https://blog.coinfabrik.com/on-ethereum-performance-evaluation-using-PoA/.

[30] "Remix," 2020, https://remix.ethereum.org/#optimize=false&evmVersion=null/.