Smart Contracts for Internet of Vehicles Car Charging Chains

Author: Du Wenke and Chen Shi Unit: Renmin University of China

Summary

In recent years, the Internet of Vehicles industry and blockchain technology development has made significant progress. With the development of technology, applications in the Internet of Vehicles industry have gradually become more popular, including intelligent navigation, autonomous driving, shared travel, etc. Blockchain technology has also received more and more attention and is widely used in the Internet of Vehicles industry to support the security and reliability of Internet of Vehicles systems. Blockchain technology can effectively improve the safety of the Internet of Vehicles system through distributed data storage and consensus mechanisms, helping users track and control the flow of data. In addition, blockchain technology can also be used to support vehicle information sharing in the Internet of Vehicles industry to achieve more efficient information exchange and data sharing.

This article aims to study smart contracts for the Internet of Vehicles car chain. First, this article analyzes the characteristics of the Internet of Vehicles car chain and common feasible solutions for smart contracts based on blockchain. It introduces many industries, including car companies and retail. actual cases. Next, this article details the application logic and solutions of the Internet of Vehicles car chain based on smart contracts. It combines the innovative projects that the team members participated in early to target the existing IoV field and combine the characteristics of the alliance chain to introduce V2G security from multiple chains. The certification scheme introduces in detail the idea scheme design and performance test design in electric vehicle network changing and tries it through the Python language of VS Code and the Solidity online editor Remix. Finally, we make an outlook and data application scenarios based on the business model and make relevant suggestions.

Keywords:

Internet of Vehicles car chain; identification and authentication; V2G; alliance chain

1. Research background

This study focuses on innovations in the latest cutting-edge technology fields, specifically the application of blockchain technology in the Internet of Vehicles industry. Since the three leading technologies involved, the Internet of Vehicles, blockchain technology, and cryptography technology, are all fields with high technical content and relatively popular industries in recent years, the background is first stated as follows.

2. Research content

(1) Characteristics of the connected car chain

The main features of the Internet of Vehicles car chain include the following six points:

1. Decentralization: The automobile chain network is a decentralized network composed of participants such as automobile companies, governments, maintenance service providers, consumers, and engine manufacturers, which can realize the safe sharing and exchange of automobile data.

2. Safe and reliable: The Internet of Vehicles car chain uses distributed accounting technology and smart contracts, which can effectively protect user data, prevent data leakage, and improve data security.

3. Data traceability: AutoChain’s blockchain technology can achieve reliable data traceability, accurately trace all aspects of the car, and help prevent car fraud.

4. Automobile data security: The Internet of Vehicles car chain can realize encryption processing of automobile data, effectively prevent the leakage of automobile data, and protect the privacy of automobile users.

5. Automatic execution: Smart contracts can be automatically executed, are safer and more reliable, and can quickly and effectively complete transactions in the automotive industry.

6. Scalability: The Internet of Vehicles car chain is scalable and can continue to develop new functions and services according to the automotive industry's needs to meet those needs.

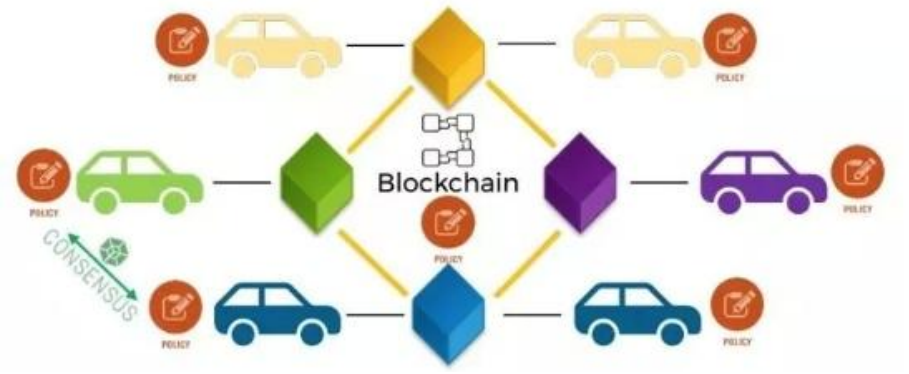


Figure 3: Combination model of Internet of Vehicles and blockchain

In terms of research in this field, many companies have been relatively successful, including ROAD, the official website (https://roadpro.io/), one incubated by INT Chian (domestic Internet of Things public chain), and many well-known Alibaba It is an Internet of Vehicles blockchain project jointly initiated by entrepreneurs. They intend to create a distributed automotive ledger system based on blockchain technology, build a ROAD Internet of Vehicles and smart transportation ecosystem with individual interaction and cluster intelligence, realize relevant business applications, and use blockchain to verify user data: Rights and privacy protection.

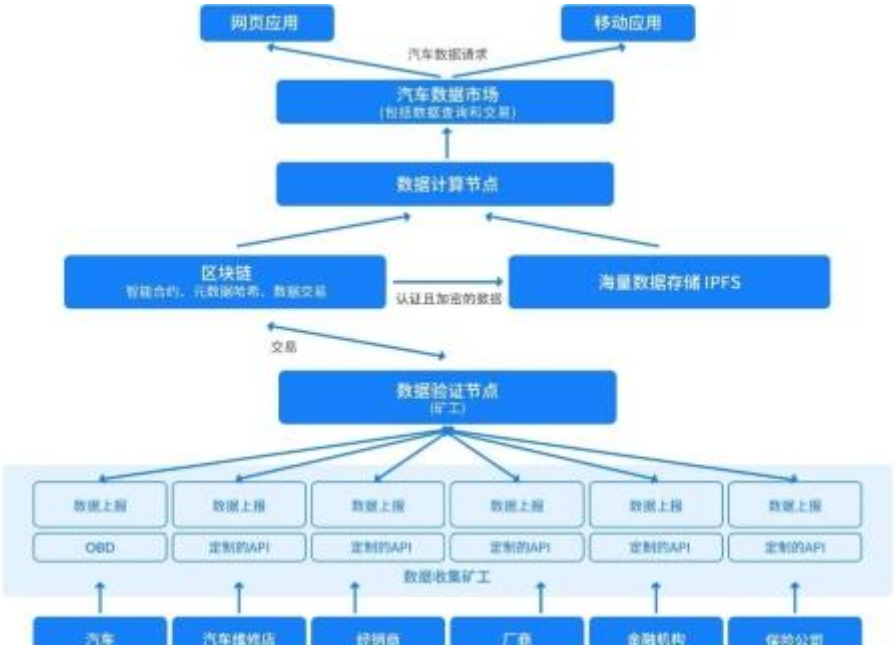


Figure 4: ROADstructural pattern

This project's first non-sensitivity-safe OBD device has been officially launched

recently. It relies on vehicle OBD interface data, uses blockchain distributed ledger and innovative contract technology to upload data to the chain, and uses desensitization algorithms to achieve privacy. Protect. Standardize and package the vehicle's condition data, driving conditions, maintenance, insurance records, etc., on the chain, and realize the value conversion after the data is uploaded to the chain, creating an innovative green ecological mine that "mines while driving."

In summary, the Internet of Vehicles industry will combine blockchain to solve existing bottlenecks and achieve privacy data protection in the future. This is the general trend of industry development.

(2) Common feasible solutions

(1) Four steps of smart contract solutions:

Commonly available solutions in the field of standard automotive chain and Internet of Vehicles include the following four steps:

Step 1: Establish a smart contract platform in the Internet of Vehicles car chain. The platform can provide a smart contract system for the automotive chain and can realize almost all business scenarios in the Internet of Vehicles automotive chain, including business processes such as transactions, payments, bills, and settlements.

Step 2: Build a smart contract development framework so that developers can quickly develop appropriate smart contract programs based on different business scenarios. The framework should include a simple, smart contract language and an executable environment to implement smart contract programs.

Step 3: Detect the security of smart contract programs running in the Internet of Vehicles car chain. To this end, a system for detecting potential vulnerabilities and security risks in smart contract programs must be developed to ensure the security of smart contract program operations.

Step 4: Specific selection of smart contract technology, such as "encrypted currency lock authentication" technology. This technology can lock the smart contract program on the blockchain to ensure its credibility.

(2)Existing solution cases:

Take McDonald's, for example, which has a fleet of more than 1,000 vehicles

and will use blockchain technology to improve its vehicle maintenance procedures. The system will use blockchain technology to track each vehicle's safety, maintenance status, and fuel efficiency and issue alerts when necessary. In addition, McDonald's uses blockchain technology to manage access control of its vehicles to ensure that only authorized drivers can drive company vehicles. The design plan is divided into PKI(digital certificate)and two identity-based schemes.

A.PKI scheme:

Solution design: Each vehicle will have private and public keys. The private key is only accessible to the vehicle's owner, while the public key is exposed to other participants. When a car uploads information such as safety, maintenance status, and fuel efficiency, it is digitally signed using the private key, and the information and signature are stored on the blockchain. Other participants can use the public key to verify the signature, ensuring the authenticity and integrity of the data. Specific steps are as follows:

The car owner or driver applies for a digital certificate from McDonald's Corporation.McDonald's Corporation issues digital certificates to car owners or drivers.Each car's digital certificate contains a public key and a private key. The private key is owned only by the car owner or driver and is used for digital signatures and authentication.

When the car needs to send data, the private key is used to sign the data to ensure the integrity of the data and the authenticity of the identity.The recipient uses the public key to verify the authenticity of the digital signature and identity information.Blockchain technology records digital certificates' issuance and verification process to ensure data security and traceability.

Performance test: McDonald's also conducted a relatively complete performance test based on the above solution: digital signature and verification are implemented using the ECDSA algorithm. In the test, a computer equipped with an Intel Core i7-8700K processor and 16GB of RAM was used, and the test results were tested using code written in Python. The test results showed that approximately 10,000 digital signature and verification operations can be performed per second.

B. Identity-based scheme:

Solution design: Each vehicle has a unique identity identifier (ID). When a car uploads information, it must first pass identity verification to ensure that the vehicle uploading the data is indeed authorized. Authentication can be achieved using digital certificates or other methods. Specific steps are as follows:

The car owner or driver registers a digital identity with McDonald's Corporation.McDonald's authenticates the car owner or driver and issues a digital identity. Digital identity includes car and driver identity information and digital certificates.When the car needs to send data, it uses a digital identity to sign it to ensure its integrity and authenticity. The recipient uses digital identity to verify the authenticity of the digital signature and identity information.

Performance test: McDonald's uses digital certificates to implement corresponding digital certificate verification. Digital certificates can be issued using public critical infrastructure (PKI) or other methods to ensure the authenticity and integrity of the certificate. In the test, a computer equipped with an Intel Core i7-8700K processor and 16GB RAM was used, and the test results were tested using code written in Python. The test results showed that approximately 100,000 digital certificate verification operations can be performed per second.

3. Solution design and performance analysis

(1) Scheme design

1. System model

In studying the existing literature, we found that the traditional PKI cross-domain authentication scheme based on PKI (Public Key Infrastructure) has single points of failure, complex certificate verification and management, and low authentication efficiency, and is not suitable for delay requirements. High V2G complex communication environment. The reference materials for writing this part come from an entrepreneurial project that member Du Wenke was fortunate enough to participate in before. Due to space limitations and the main purpose of this article is to sort out the application of existing blockchain technology in the Internet of Vehicles industry,

it is only briefly mentioned here. Describe this V2G security authentication scheme based on master-slave multi-chain.

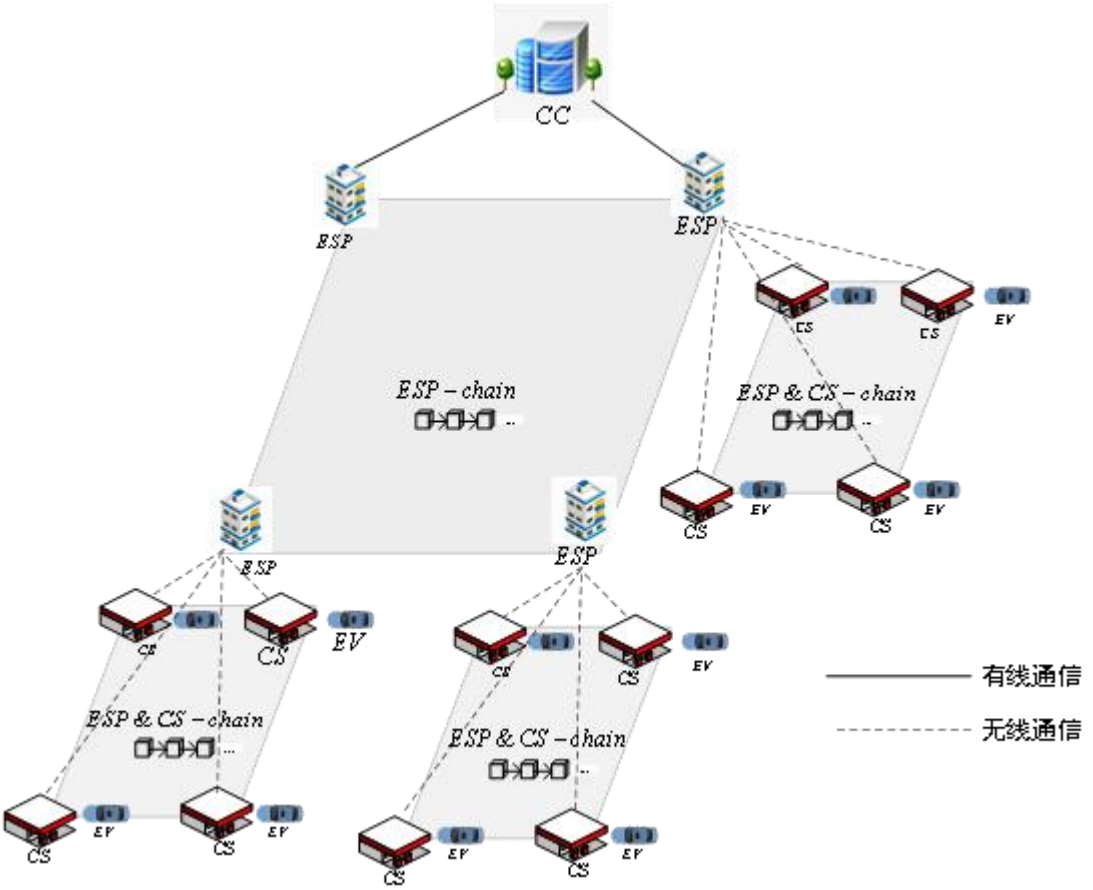


Figure 5: V2G certification scheme structure diagram

In the above system model:

(1) Electric power service provider ESP: It is the electric power service provider in a certain area and is the core of the system. It includes a trusted center (TA) and a convergence unit. The trusted center is responsible for generating public and private keys for charging stations and registered electric vehicle users, issuing certificates for them, and registering the identity information of legal participants. The aggregation unit gathers charging stations in a certain area and is responsible for the coordinated control of electric vehicle charging in the area and controls the flow of electric energy in the power grid. And participate in maintaining the data chain and certification chain at the same time.

(2) Electric vehicles EV: Electric vehicles are the main participants in the entire system and have bidirectional chargers that can perform charging and discharging

operations.

(3) Charging station CS: The charging station participates in the maintenance and accounting of the data link. Each charging station is equipped with multiple charging stations and can charge multiple EVs in parallel.

(4) Blockchain network BC: This solution adopts a multi-layer blockchain structure and a certification chain composed of various ESPs. The certification chain is used to store the system public key to facilitate subsequent charging of electric vehicle users across service providers. safety certificate. The data storage chain is composed of each ESP and the charging station CS within its range, and stores electric vehicle registration information. Both the data chain and the certification chain are based on the alliance chain and allow audited nodes to join.

2. Introduction to alliance chain

The existing blockchain is divided into three parts: private chain, public chain and alliance chain, and new forms may be developed in the future. This plan mainly relies on the alliance chain to develop the design.

A consortium chain is a cluster composed of multiple private chains, a blockchain managed by multiple institutions. Each organization or institution manages one or more nodes, and its data is only allowed to be read, written and sent by different institutions within the system. . Each node of the alliance chain usually has a corresponding entity organization, and can only join and exit the network after being authorized. Various institutions and organizations form an alliance of stakeholders to jointly maintain the healthy operation of the blockchain.

Alliance chain features: including

1） Partial decentralization is different from the public chain. To some extent, the alliance chain is only owned by the members within the alliance, and it is easy to reach a consensus, because after all, the number of nodes in the alliance chain is very limited.

2） The public chain has strong controllability. Once the blockchain is formed, it cannot be tampered with. This is mainly due to the fact that the nodes of the public chain are generally massive. For example, there are too many nodes

in Bitcoin. It is almost impossible to tamper with the block data. In the alliance chain, as long as the majority of all institutions reach a consensus, the block data can be changed;

3） Data will not be made public by default. Unlike public chains, the data of the alliance chain is only accessible to institutions and users in the alliance.

4） The transaction speed is very fast. Just like the private chain, the alliance chain is essentially a private chain. Therefore, due to the small number of nodes, it is easy to reach consensus and the transaction speed is naturally much faster.

2. System initialization

ESP_i Select the cyclic group G with prime order n, select the elliptic curve point P on the finite field Fq, and use it as the generator of the group G. ESP_i Choose a secure one-way hash function $h_1:\{0,1\}^* \to G$. ESP_i Choose a random number $s_i \in Z_n^*$ As the system master private key, the system public key is calculated $PK_{ESP_i} = s_i \cdot P$, ExpT_{EV_n} Keep secret s_i And expose system parameters: $params_i = \{G, n, P, h_1, h_2, PK_{ESP_i}\}$.

Each ESP is pre-certified as a blockchain node. First, use the account generation tool to generate a blockchain account for each node, and initialize the blockchain network. During the initialization process, a genesis block is generated, and subsequent blocks are linked to the newly generated Blocks form a blockchain. Publish the pre-designed smart contract to the blockchain network, each node records the contract address and application program interface (API), and then interacts through the smart contract address and interface to trigger the execution of the smart contract.

3. Registration stage

All EVs and CSs participating in the V2G network need to register their identity information with the ESP, and the communication channel between the ESP and them is secure.

EV_{ij} register: EV_{ij} First of all, you need to ESP_i To initiate a registration request,

the user submits information such as car license plate, driver's ID number, phone number, etc. $\mathrm{ESP_i}$ IDs will be generated for approved users. $ID_{EV_{ij}}$, $\mathrm{ESP_i}$ passECCAlgorithm assigns public and private keys to it $(\mathrm{PK_{EV_{ij}}}, \mathrm{SK_{EV_{ij}}})$, and the validity period $\mathrm{ExpT_{EV_{ij}}}$, ESP_1 for $\mathrm{EV_{ij}}$ Generate certificate: $\mathrm{Cert_{EV_{ij}}} = < \mathrm{ID_{EV_{ij}}}, \mathrm{PK_{EV_{ij}}}, \mathrm{ExpT_{EV_{ij}}}, \mathrm{Sig_{ESP_{EPij}}} >$

in $\mathrm{ExpT_{EV_{ij}}}$ is the validity of the certificate, $\mathrm{Sig_{ESP_{EPij}}}$ yes $\mathrm{ESP_i}$ right $\mathrm{ID_{EV_{ij}}}$、$\mathrm{PK_{EV_{ij}}}$、$\mathrm{ExpT_{EV_{ij}}}$ signature, if $\mathrm{EV_{ij}}$ registration success, $\mathrm{ESP_i}$ Will $\{ID_{EV_{ij}}, PK_{EV_{ij}}, ExpT_{EV_{ij}}\}$ Encapsulate it into a transaction and publish it to the blockchain network, triggering the smart contract registration algorithm. After the consensus transaction of each node is successfully verified, the user EV_{ij} The registration information will be added to the data storage chain, and now all join the local domainV2Gnetwork EV_{ij} The registration information is saved in the ESP_i within its scopeCSOn a jointly maintained data storage chain. Finally, through a safe channel, $\{ID_{EV_{ij}}, PK_{EV_{ij}}, ExpT_{EV_{ij}}, SK_{EV_{ij}}, Cert_{ESP_{EPij}}\}$ Sent back to registered vehicle $ID_{EV_{ij}}$, users keep their private keys, certificates and other information privately.

Same method, ESP_i to which it belongs CS_{ij} Generate public and private keys $(PK_{CS_{ij}}, SK_{CS_{ij}})$ and certificate $Cert = \{ID_{CS_{ij}}, PK_{CS_{ij}}, ExpT_{CS_{ij}}, Sig_{ESP_{CSij}}\}$.

4. Blockchain generation

(1) Certification chain generation

After the certification chain network is initialized, the system parameters generated by itself are uploaded to the certification chain, which include system public keys and other parameters. They need to go through distributed public knowledge and the verified information can finally be written into the blockchain. Encapsulate the system parameters into a transaction, sign the transaction with its private key, and then publish the transaction to the blockchain network. After receiving the transaction, other nodes first verify the validity of the transaction. If the

node receives the transaction and passes the verification,

Forward its verification results to other verification nodes in the certification chain, and write the transaction into the transaction pool. Afterwards, the master node is selected, and the master node acts as an accounting node to package the transactions of the transaction pool within the time into a new block, and broadcasts the block to other nodes in the blockchain network, and other nodes verify the block after receiving it. The validity of the verification results is then broadcast to the entire network. After receiving the verification results, each node compares its own verification results with those of other nodes and feeds the results back to the master node. If the master node receives the node's consent, the validity of the block is verified and added to the certified blockchain as a new block.

(2) Generation of data link

The first step is to prepare the data before uploading it to the chain: digitally sign the data to be uploaded to the chain to ensure the integrity of the data and the authenticity of the identity. At the same time, the data needs to be formatted to facilitate the storage and query of the blockchain.

The second step is to encapsulate the data into a transaction: encapsulate the digitally signed and formatted data into a transaction, and use the private key to sign the transaction to obtain a summary of the transaction. The digest contains the content and signature information of the transaction, ensuring the security and credibility of the data.

The third step is to broadcast the transaction to the network: the transaction is sent to the blockchain network to wait for verification and confirmation. Other nodes will verify the validity of the transaction and write it into the transaction pool to wait for confirmation.

The fourth step is for the master node to package: the master node will regularly select some transactions from the transaction pool for packaging and generate new blocks. During the packaging process, the masternode needs to verify the validity and correctness of each transaction before combining them into a new block.

The final step is block verification and confirmation: other nodes verify the

validity and correctness of the new block and feed the results back to the master node. If the masternode receives enough consent, the validity of the block is verified and added to the blockchain

4. Mutual authentication stage

Electric vehicle users EV_{ij} come CS_{ij} Nearby, want to addEnter V2GCharge and discharge transactions are carried out on the network. first EV_{ij} Towards CS_{ij} Send your own access request information $\{M_{req}, ID_{EV_{ij}}, PK_{EV_{ij}}, T_{EV_{ij}}, Cert_{EV_{ij}}, Sig_{EV_{ij}}\}$,in M_{req} for EV_{ij} charging request information, $T_{EV_{ij}}$ is the timestamp of this message, $Sig_{EV_{ij}}$ for EV_{ij} PickUse ECDSASignature algorithm for messages $\{ID_{EV_{ij}}, T_{EV_{ij}}, M_{req}\}$'s signature.

The specific steps for signing are as follows:

（1）first EV Randomly select an integer k, $k \in [1, n-1]$, and calculate $k \cdot G = (x, y)$, $r = x \bmod n$;

Calculate again $H = h_1(ID_{EV} \| T_{EV} \| M_{req})$, $s = (H + r \cdot SK_{EV}) \cdot k^{-1} \bmod n$, then the signature is

$Sig_{EV} = (r, s)$;

（2）EV will $r' = r$ Sent to CS via secure channel. CAfter S receives the message sent by EV, it first verifies the timestamp. T_{EV} Is it valid? After verification, query the EV registration information through the storage chain, verify the validity of the EV certificate, and complete the verification of the signed message. The verification steps of the signed message are as follows:

1) CScalculate $H' = h_1(ID_{EV} \| T_{EV} \| M_{req})$, $u = s^{-1} \cdot H' \bmod n$, $v = s^{-1} \cdot r \bmod n$,CSQue ry from the chainEV registration information, extract EVThe public key of $(x', y') = uG + vPK_{EV}$,and $r' = x' \bmod n$. If the equation $r' = r$ is established, thenCS finish vs. EVcertification.

（3）CSWill EVThe charging request is sent to ESP, through ESPMake EVParticipate in V2Gnetwork to formulate optimal charging strategies for electric vehicles in each charging station. CSTo ESPForward EVcharging request: $\{M_{req}, PK_{CS}, ID_{CS}, T_{CS}, Cert_{CS}, Sig_{CS}\}$,in T_{CS} is the timestamp of this message, Sig_{CS} Yes CS adopts ECDSAAlgorithm to message $\{M_{req}, ID_{CS}, T_{CS}\}$'s signature.

（4）ESPReceived CSAfter the message, first verify the timestamp T_{CS} Is thereis valid, then verify the certificate of CS and complete the verification of the signed message. If the verification passes, the certification of CS by ESP is completed..

（5）ESPElectric vehicles will be allowed to participate in V2Gnetworkinformation M_{res} Send back to CS: $\{M_{res}, ID_{ESP}, PK_{ESP}, T_{ESP}, Sig_{ESP}\}$,in T_{ESP} is the timestamp of this message, Sig_{ESP} yesESPto news $\{M_{res}, ID_{ESP}, T_{ESP}\}$'s signature.

（6）CSAfter receiving the message, first verify the timestamp. After the verification is valid, use ESPThe public key of C verifies the signed message. If the verification passes, then CSreceiveESPmessage, completed CSto ESPverification.

（7）CSTo EVReturn successfully join V2GNews from the web: $\{M_{res}, ID_{CS}, T'_{CS}, PK_{CS}, Cert_{CS}, Sig'_{CS}\}$,in T'_{CS} is the timestamp of this message, Sig'_{CS} Message for CS $\{M_{res}, ID_{CS}, T'_{CS}\}$'s signature. After EV receives the message, it first verifies the timestamp and the verification is valid.After that, check C againSpublic key certificate and use CSThe public key of E verifies the correctness of the signed message. If the verification passes, EVAccept the message, EVJust come to the designated charging station to charge.

5. Smart Contract

a. Four-step contract design

(1) Registration contractCcontractRregister

The first step of the contract, the format of Attribute attributes:[role:ordinary,

tag:A]

1) Input{vehicle ID, tag, attributes} triggers the registration function registerPK() in the smart contract
2) The key generation algorithm uses ECDSA to generate public and private keys (SK, PK), where PK is the public key and SK is the private key.
3) Send the generated {SK, ID, Cert, tag} to the registered vehicle terminal EV (Cert is the ESP's signature for the message {PK, ID, tag, T})
4) Store {ID, PK, tag, time} on the chain in the form of a transaction. time is the time when the key is generated

First, we need to define a structure to represent vehicle information and the corresponding public key:

```
1.struct Vehicle {
2. stringVIN; //Vehicle VIN code
3. stringpublicKey; //Public key of the vehicle
4.}
```

Next, we define a mapping type state variable to map each VIN code to the corresponding vehicle information:

```
1.mapping(string => Vehicle) publicvehicles;
```

Then, we define a registration function to let the vehicle register its own VIN code and public key:

```
1.function register(stringmemory_VIN, stringmemory_publicKey) public {
2. // Check if the VIN code has been registered
3. require(bytes(vehicles[_VIN].VIN).length== 0, "Vehicle already registered");
5. //Create a new vehicle structure and add it to the mapping
6. Vehiclememory newVehicle= Vehicle(_VIN,_publicKey);
7.vehicles[_VIN] =newVehicle;
8.}
```

In the registration function, we first use the require statement to check whether the VIN code has been registered. If the VIN number already exists, an exception is thrown. If the VIN is not registered, we create a new vehicle structure and add it to the mapping.

Result example graph:

1) The system log of the ESP during EV registration is as shown in the figure.

You can see that the contract on the side of the vehicle when registering triggered an Ethereum event. You can obtain identity credentials by listening to the event.

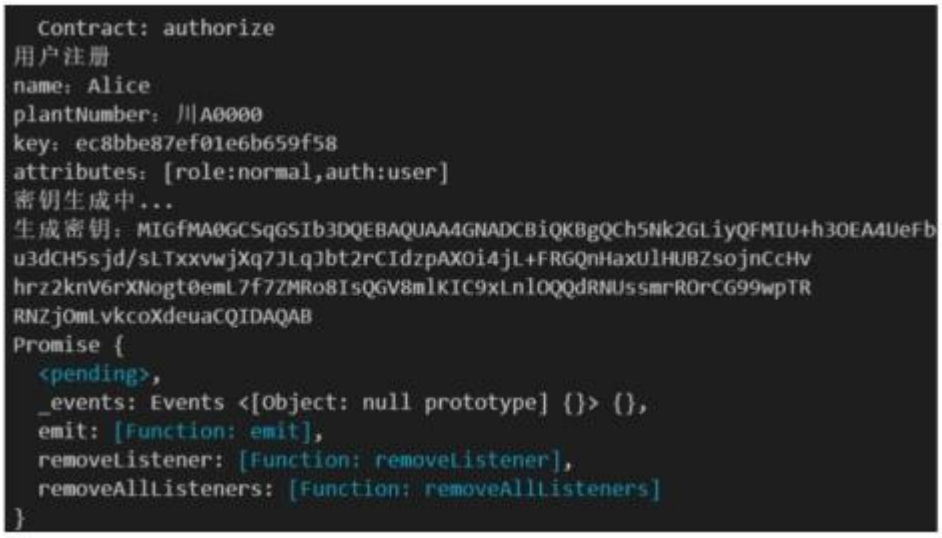
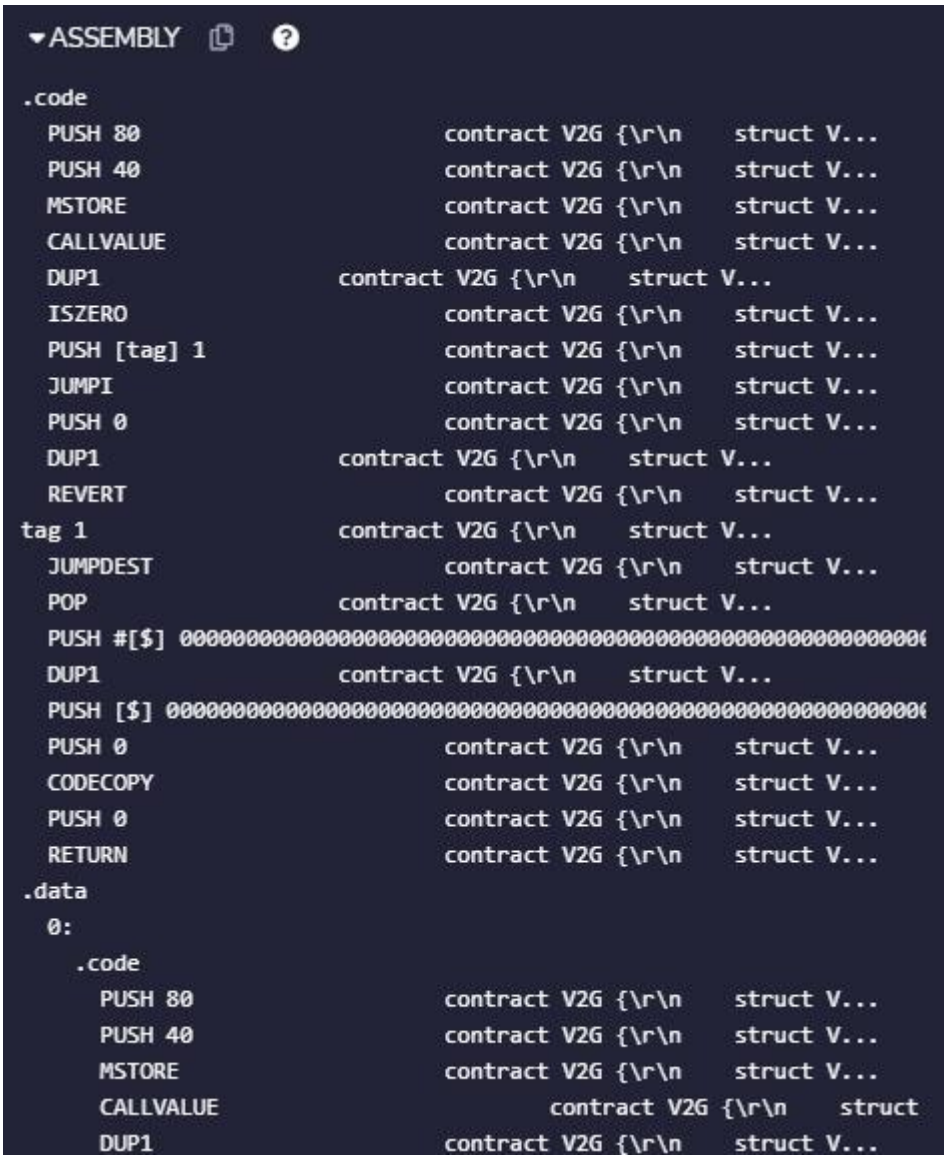


Figure 6: Schematic diagram of registration contract results

2) Listen to events to obtain identity credentials

Figure 7: Identity certificate transaction diagram

3） RemixCode screenshot



(2) contract query contract query

1) Build a key word list 2) Input{ID, tag} triggers a query to see if {EV1, A} is entered into the contract and the transaction will be automatically retrieved 3) The page displays (ID, tag, PK, time) where time is the time when the key is generated . Example: 1. The vehicle terminal will query the public key of CS or ESP; 2. The CS terminal will query the public key of EV or ESP.

Based on the hash value of Merkle Root, you can quickly determine whether the transaction is in the block. Calling Ethereum's prefix tree matching algorithm can quickly match account data. (I don’t understand that this prefix tree matching algorithm can be called during simulation only if it is written in the paper).

算法 3-2 查询合约

```
1. function querySPK (TAID) {
2. % 由 TA 调用检索特定的系统公钥
3. For(i=0;i<SPK.length;i++)
4. if Exist(SPK[i].TAID== TAID) then
5. return SPK[i];
6. else;
7. return 0;
8. }
9. function queryVPK (ID) {
10. % 由 TA 调用检索特定的车辆公钥
11. For(i=0;i<VPK.length;i++)
12. if Exist(VPK[i].ID == ID) then
13. return VPK[i];
14. else;
15. return 0;
16. }
```

Figure 8: Schematic of querying contract code

```
1.pragma solidity^0.8.0;
2.
3.contractChargingRecords {
4.
5. struct ChargingRecord {
6.address vehicleAddress;
7. uintstartTime;
8. uintendTime;
9. uint 能源;
10. }
11.
12.mapping(address=> ChargingRecord[]) privatechargingRecords;
13.
14. functionaddChargingRecord(address_vehicleAddress, uint_startTime, uint_endTime, uint_能源) public {
15.chargingRecords[_vehicleAddress].push(ChargingRecord(_vehicleAddress,_startTime,_endTime,_能源));
16. }
17.
18. functiongetChargingRecordCount(address_vehicleAddress) publicview returns(uint) {
19. returnchargingRecords[_vehicleAddress].length;
20. }
twenty one.
twenty two. functiongetChargingRecord(address_vehicleAddress, uint_index) publicview returns(address, uint, uint,
uint) {
```

```
twenty three. ChargingRecordmemory record=chargingRecords[_vehicleAddress][_index];
twenty four. return (record.vehicleAddress,record.startTime,record.endTime,record.能源);
25. }
26.}
```

This contract is used to store charging records of electric vehicles. Each charging record includes the vehicle address, charging start time, charging end time and charging power. The contract includes the following methods:

addChargingRecord: used to add new charging records to the contract.

getChargingRecordCount: used to obtain the number of charging records of a specific vehicle.

getChargingRecord: used to obtain the specific charging record of a specific vehicle.

Figure 8: Query contract RemixRun screenshot

(3) Contract revoke

Call revokePK(). When the vehicle's reputation value is low or malicious behavior occurs, the illegal EV key will be revoked from the blockchain based on the ID, or the authorization to the third party will be revoked.

```
1.pragma solidity^0.8.0;
2.
3.contractAuthorization {
4.
5.mapping(address=>mapping(address=> bool)) privateauthorized;
6.
7. event RevokeAuthorization(address indexed vehicleAddress,address indexed thirdParty);
8.
9. functionauthorize(address_vehicleAddress,address_thirdParty) public {
10.authorized[_vehicleAddress][_thirdParty] = true;
```

```
11. }
12.
13. functionrevokeAuthorization(address_thirdParty) public {
14.authorized[msg.sender][_thirdParty] = false;
15.emitRevokeAuthorization(msg.sender,_thirdParty);
16. }
17.
18. functionisAuthorized(address_vehicleAddress,address_thirdParty) publicview returns(bool) {
19. returnauthorized[_vehicleAddress][_thirdParty];
20. }
twenty one.}
```

The contract includes the following methods:

authorize: used to authorize third parties to access vehicle information.

revokeAuthorization: Used to revoke access rights previously given to third parties.

isAuthorized: used to check access rights between a specific vehicle and a specific third party

(4) update contract

Since the vehicle public key is updated regularly, the process of updating the user public key is discussed here. When EVWhen the current public key is about to expire, the user needs to report to ESPApply to obtain a new public and private key so that you can continue to enjoy V in the future2Gcharging services to reduce the number of certifications. The update process of user keys is similar to the registration process, EVTo ESPSend a request for key update $M_{up} = \{PK_{gv}, ID_{gv}, ExpT_{gv}\}$,ESPAfter receiving the key update request, it will be reviewed. If the verification passes, it will be EVGenerate new public and private keys $(newPK_{gv}, newSK_{gv})$ and certificate $Cert'_{gv} = < ID_{gv}, newPK_{gv}, ExpT'_{gv}, Sig_{ESP_{cr}} >$ .ESP will update the publicThe key information is encapsulated into a transaction, and the node executes the registration function in the contract EV_{Ai} , the transaction is successfully executed in, and will be added to the blockchain after being verified by other nodes and reaching a consensus, and returnedGive to ESPsuperiorchain results. FollowESP will be updated laterThe private key and certificate are sent back to the registered vehicle

in a secure manner, and the user key update process is completed.

(1) Blockchain simulation

The specific process of blockchain simulation is divided into the following three steps:

1） The data chain stores the public key PK and identification field tag (A, B, C...) registered by the vehicle user using the ECDSA algorithm, as well as the ID: EV1, EV2... and the public keys of ESP and CS.

2） Install the Ethereum client geth on ESP and CS to implement the blockchain node function (you can set up three nodes and three hosts: one ESP node and two CS nodes)

3） EV checks the data on the chain. When the vehicle terminal enters the system for the first time, it triggers the registration function. After establishing a connection with the ESP, it sends identity information to the ESP node. The registration contract is responsible for generating transactions containing {PK, ID, tag, T} and entering them into the data chain.

The specific execution level involves two parts: system initialization and simulation environment:

1)system initialization

A. ESP side running node software environment Ubuntu20.04, Geth1.9.9-stable, NodeJs v16.13.1, Solidity v0.7.4

B. On-board EV account software environment Ubuntu20.04, Geth1.9.9-stable, Web3.js v1.0.0, Ganache CLI v6.4.3

C. The Ganache framework generates accounts and keys.

2) Simulation environment:

A. The Truffle framework is used for contract management and deployment, and the Ganache framework is used for account and key generation. First, use the truffle init command to initialize the project directory, automatically generate the compilation and migration environment of the contract, save the written contract in the contracts folder, write the migration script in the migrations folder, modify the truffle-config.js configuration file to migrate

the contract to the deployed data link.

B. First, use Ganache to generate 3 vehicle-end EV accounts for contract call testing, and the first account is used for contract deployment. Run the truffle compile command to compile the contract and generate the binary interface file of the contract, and then execute the truffle migrate command to deploy the contract to the data chain.

Members of this section passed pytholanguage for simple simulation, by creating a simple Flaskapplication to represent nodes in a blockchain network and SSolidityOnline editorRemixTry the code, please refer to the code package for details

```
1.pragma solidity^0.8.0;
2.
3.import "./V2GAuth.sol";
4.
5.contract V2GAuthBenchmark{
6.V2GAuthprivatev2gAuth;
7.
8.uint256privatenumVehicles;
9.uint256privatenumUpdates;
10.
11.uint256privatestartTime;
12.uint256privateendTime;
13.
14. functionsetup
15.
```

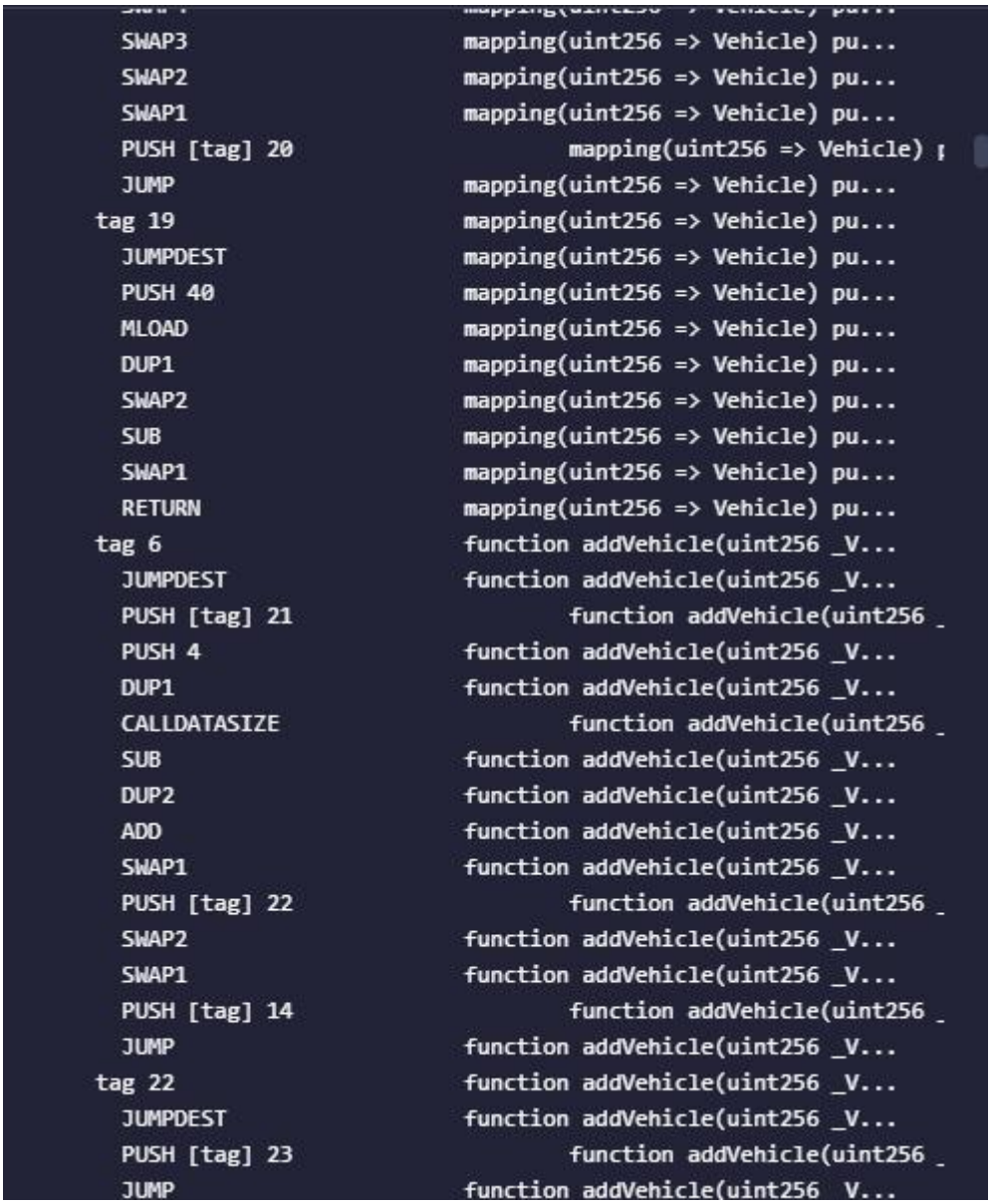


Figure: Screenshot of performance test run

（二） Blockchain performance analysis

We mainly complete the performance analysis of the blockchain through the following four indicators, namely storage consumption, query efficiency, delay analysis and transaction throughput.

1. Storage consumption

By analyzing the smart contract execution of each node's EVM and how many bytes the block header occupies, estimate how many bytes the average transaction on the blockchain occupies, and how many transactions does a block contain on average? (Is it fixed or can it be set in the genesis block configuration file)?

2. Query efficiency

The total transaction volume is 1000~5000, and the query delay is incremented by 500 each time (the time to query the public key of an entity). The abscissa is the total transaction volume, and the ordinate is the time when the block was queried.

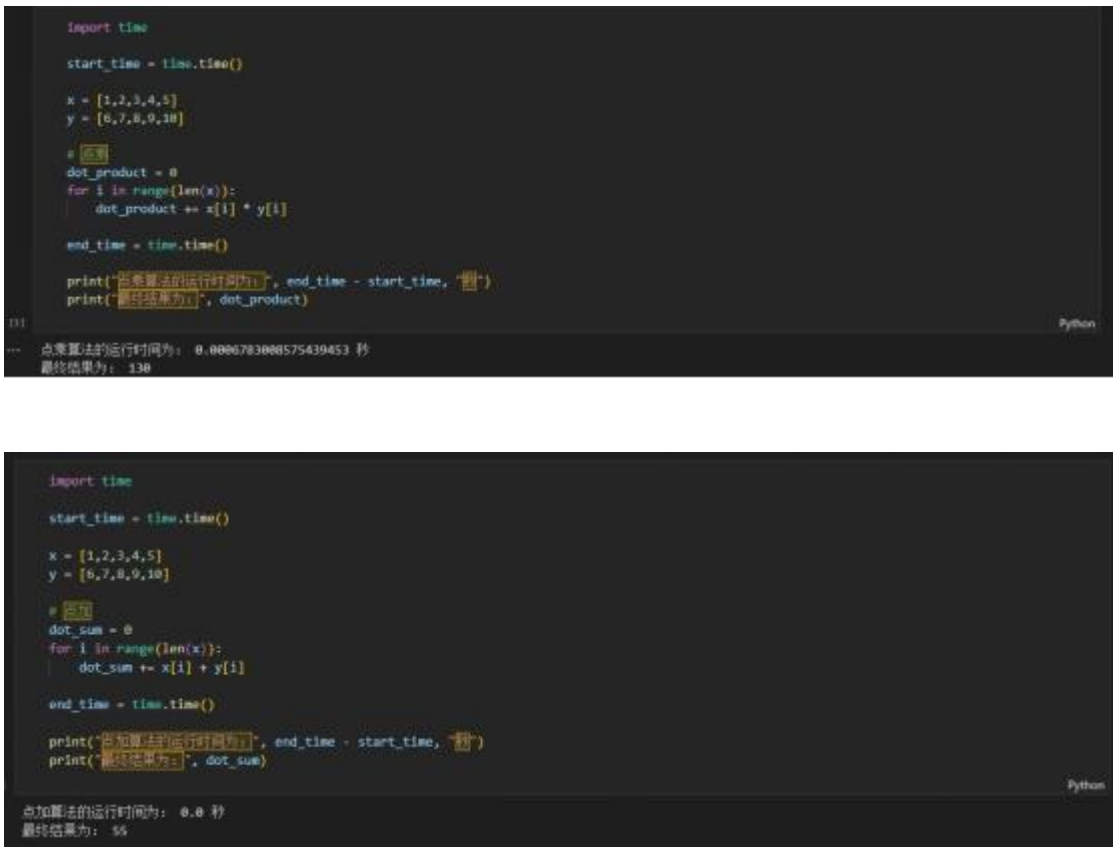


Figure: Query efficiency test screenshot

3. Delay analysis

The transaction on-chain delay refers to the time it takes for a transaction to be created, packaged into a block by the miner node, and successfully uploaded to the chain through consensus. In Ethereum, the mining difficulty can be changed by adjusting the difficulty parameter of the genesis block. Therefore, we used two CPU configurations to measure the transaction on-chain delays with different difficulty parameters and adjusted the difficulty value to test the delay of block packaging.

difficulty 参数 \ 区块封装时间 (ms)	CPU1	CPU2
0X5FFFFA	11967	14256
0X99999	1152.09	1198.99
0X50000	564.13	589.32
0X40000	431.15	491.42
0X30000	299.11	311.34
0X20000	277.56	309.87
0X10000	241.53	272.94

Figure 8: Screenshot of delay test example

Figure: Query latency test screenshot

4. transaction throughput

We record the transaction acceptance volume and transaction processing volume through the coordinate system (transaction acceptance volume per second on the abscissa; transaction processing volume per second on the ordinate)

Example: 500 cars send registration requests to ESP to generate keys at the same time in one second. See how many of the public keys of these 500 cars are packed into the block in one second. Then change the abscissa to the total number of

transactions per second, and then test the transaction processing volume per second to form a comparison chart.

```python
import time
from threading import Thread
from queue import Queue
import requests

def test_throughput(num_transactions, num_threads):
    blockchain = Blockchain()
    threads = []
    for i in range(num_threads):
        t = Thread(target=mine_blocks, args=(blockchain, num_transactions//num_threads))
        threads.append(t)

    start_time = time.time()
    for t in threads:
        t.start()

    for t in threads:
        t.join()

    end_time = time.time()

    total_time = end_time - start_time
    throughput = num_transactions / total_time

    print("Number of Transactions: {}".format(num_transactions))
    print("Number of Threads: {}".format(num_threads))
    print("Total Time: {} seconds".format(total_time))
    print("Throughput: {} transactions per second".format(throughput))

def mine_blocks(blockchain, num_transactions):
    for i in range(num_transactions):
        transaction = {"sender": "Alice", "recipient": "Bob", "amount": 1.0}
        blockchain.add_transaction(transaction)
        blockchain.mine_block()

if __name__ == '__main__':
    test_throughput(num_transactions=1000, num_threads=10)
```

```
Number of Transactions: 1000
Number of Threads: 10
Total Time: 0.009334087371826172 seconds
Throughput: 107134.2017879949 transactions per second
```

Figure: Screenshot of transaction throughput test

Winding

The team members in this section tried to design an on-chain solution, but since most of them require payment, they just did a local simulation.

In the code, the generate_key_pair function is first used to generate an RSA key pair, and then the sign_data and verify_signature functions are defined to sign the data

and verify the signature. The store_to_blockchain function is used to store public keys, data and signatures on the blockchain.

In the main function of the code, we define the user information and timestamp and merge them into a byte string. The data is then signed using the private key, and the signature is verified using the public key. If the signature verification is successful, the store_to_blockchain function is called to store the public key, data and signature on the blockchain.

```
1. from Crypto.PublicKey importECC
2.from Crypto.Hash importSHA256
3.importtime
4.
5.#Generate ECDSA key pair
6.key=ECC.generate(curve='secp256k1')
7.public_key=key.public_key().export_key(format='PEM')
8.
9.# Prepare data for blockchain
10.vehicle_id= 'ABC123'
11.timestamp= int(time.time())
12.data= {'public_key':public_key, 'vehicle_id':vehicle_id, 'timestamp':timestamp}
13.
14.# Compute SHA-256 hash of data
15.h=SHA256.new(str(data).encode()).hexdigest()
16.
17.# Store data in blockchain
18.blockchain= [{'data':data, 'hash':h}]
19.print(blockchain)
```

Data application

The team members in this section focus on the alliance chain participants, which are centered on power suppliers, and other players include charging piles and charging cars. This solution is also applicable to oil- and gas-burning cars, so it can be widely used in vehicle information sharing. In terms of data applications,

1. For car owners: In view of the shortcomings of traditional cars such as manual confirmation of charging balance and charging time, through uplink transmission, it helps car owners and charging stations automatically update data in milliseconds, and car owners only need to

complete the last step of picking up the car.

2. For car owners: The data recorded by charging piles such as pulse and efficiency during the charging process can indirectly reflect the quality, potential scrapping risk and lifespan of the vehicle. It can remind car owners of maintenance-related matters as early as possible, further realizing information exchange between people and vehicles.
3. For the government: Based on the differential distribution of charging pile usage frequencies in a specific area, government agencies can better grasp urban traffic flow and other positioning conditions, and further improve the implementation timeliness of infrastructure construction - such as unscientific geographical location design Charging piles (too idle) move to tight supply and demand areas
4. For the government: In addition, the government can also judge the activity level of new energy vehicles and traditional vehicles by detecting the use of charging piles in different districts of a city, and make further decisions on intelligent emission reduction work in the district. Instructions and adjustments
5. For car companies: if they obtain vehicle usage data after obtaining the permission of the three parties involved in the alliance chain, it can help them better improve vehicle performance (such as discharge efficiency and safety of use, etc.)

(1) chapter summary

In summary, since there are basically only one or a few power suppliers in a region (the monopoly nature of the power grid industry), the alliance chain of this solution can perform well differentiated analysis through the design with power suppliers as the core. Although this solution only focuses on the charging of shared cars or the Internet of Vehicles, similar alliance chain structure design can be applied to multiple links: such as traffic lights to judge and record violations. Through such a design, vehicles can be Information is encrypted and shared among car owners, car companies, and governments, ultimately realizing the information interoperability

model that the Internet of Vehicles industry has long expected.

4. Future development

The application of smart contracts in vehicle network security, privacy and location identification is a major tool for the development of the automotive industry. It can not only improve the safety of vehicles, but also meet the privacy needs of consumers. The business model and profit model of smart contracts mainly revolve around the three aspects of vehicle safety, privacy and location identification, specifically including:

First is the safety aspect. Smart contracts can enable automobiles to have a higher level of security. In terms of automobile safety, smart contracts can be used to conduct security detection, detect vehicle safety hazards, and implement security reinforcement of the in-vehicle network. In addition, smart contract technology can also be used to achieve real-time positioning of vehicles and remote control of vehicles.

Second is the privacy aspect. Smart contracts can protect the privacy of automobiles and vehicles, prevent vehicle data from being stolen by third parties, and protect the driving tracks of automobiles and vehicles to prevent theft and theft. In addition, smart contracts can also be used to identify vehicles and prevent vehicle intrusion and theft.

Finally, there is the aspect of location recognition. Smart contracts can realize real-time tracking of vehicle locations and route planning for vehicles, thereby making cars drive safer and faster. In addition, smart contracts can also realize real-time monitoring of vehicles and vehicle behavior analysis, thereby better protecting the safety of automobiles and vehicles.

The business model and profit model of smart contract applications in vehicle network security, privacy and location identification are mainly formed from the three aspects of vehicle security, privacy and location identification. The business model is mainly based on Internet of Vehicles vehicle security protection services, vehicle network security consulting services, vehicle network security training services, etc.,

combined with smart contract technology, through vehicle safety detection, vehicle identity recognition, vehicle real-time positioning, vehicle behavior analysis and other functions , to provide vehicle safety, privacy and location identification services.

In terms of profit model, corresponding profits can be obtained by charging vehicle safety service fees, vehicle network security consulting service fees, vehicle network security training service fees, etc.

In the future, the development prospects of smart contract applications in vehicle network security, privacy and location identification are very broad. The automobile industry is constantly developing, and smart contract technology will also develop accordingly, which will bring more security, privacy and location identification services to the automobile industry, making the development of the automobile industry safer and more efficient.

Author contributions

Du Wenke: wrote the main part of the paper; Chen Shi: PPTProduction and other work. Overall, the two students communicated and supported each other during the completion of the project, cooperated with each other in a reasonable division of labor, and trusted each other.

appendix

"Automotive Big Data Operation Based on Alliance Blockchain"https://zhuanlan.zhihu.com/p/453308711

"Based on hyperledgerAlliance chain car trajectory tracing system》

https://blog.csdn.net/weixin_44062177/article/details/116401273

Blockchain simulation code package.zip

Performance test local code.ipynb

references

[1] Wang Chuanhua, Zhang Quan, Wang Huimin, Xu Xin, Ma Oubo. Internet of Vehicles reputation model with privacy protection under blockchain architecture [J]. Journal of Zhejiang University (Engineering Edition): 1-13.

[2] Liu Xuejiao, Cao Tiancong, Xia Yingjie. Research on efficient cross-domain data

security sharing of Internet of Vehicles under blockchain architecture [J]. Journal of Communications: 1-12.

[3] Liu Wei, Guo Lingbei, Xia Yujie, She Wei, Tian Zhao. Raft consensus algorithm based on credit evaluation model [J]. Computer Science: 1-9

[4] Zhang Qingqing, Tian Xiao, Tian Jin. Research on trusted identity scheme for Internet of Vehicles based on blockchain oracle [J]. Information Security Research, 2023, 9(02): 120-126.

[5] Yang Lei, Long Wei. Blockchain-based trust mechanism for Internet of Vehicles [J]. Computer Application Research: 1-8.

[6] Zheng Jianwen. Blockchain-based Internet of Vehicles digital forensics system [J]. Information Technology and Informatization, 2022, (12): 80-83.

[7] Hu Yi, She Kun. Dual-chain Internet of Vehicles system based on blockchain and smart contracts [J]. Information Network Security, 2022, 22(08): 26-35.

[8] Hu Yi. Design and implementation of smart contracts for automobile chains[D]. University of Electronic Science and Technology of China, 2022.

[9] Zhang Zhen, Meng Kexin. Innovative transportation equipment supports modern logistics-Foton Motor provides customers with smart solutions for urban logistics and transportation [J]. China Economic and Trade Guide, 2018, (24): 27-29.

[10] Li Yang. Design and application of zero-knowledge proof in car wallets[D]. University of Electronic Science and Technology of China, 2022.

[11] Lin Xudan, Bao Shijian, Zhao Lixin, Zhao Chenglin. Solution design and performance analysis of automotive supply chain system based on Hyperledger Fabric [J]. Computer Science, 2020, 47(S1): 546-551.

[12] Zhang Jing, Hu Ningyu, Feng Liping. Design and implementation of supermarket purchase, sale and inventory management system based on Java [J]. Information and Computers (Theoretical Edition), 2022, 34(18): 124-127+131.

[13] Li Baodong, Ye Chunming. Automotive supply chain product traceability system based on blockchain [J]. Computer Engineering and Applications, 2020, 56(24): 35-42.

[14] Zhang Qingqing, Tian Xiao, Tian Jin. Research on trusted identity scheme for

Internet of Vehicles based on blockchain oracle [J]. Information Security Research, 2023, 9(02): 120-126.

[15] Liu Kun, Wang Yuanjie, Shen Zihao, Wang Hui, Liu Peiqian. Trusted prediction cache architecture for privacy protection of Internet of Vehicles combined with blockchain [J]. Journal of Beijing University of Posts and Telecommunications, 2022, 45(06): 140-146.

[16] Hao Min, Ye Dongdong, Yu Rong, Wang Jingyu, Liao Jianxin. Blockchain-empowered 6G zero-trust Internet of Vehicles trusted access solution [J]. Journal of Electronics and Information, 2022, 44(09): 3004-3013.