EasyChair Preprint

№ 5918

BDIM: a Blockchain-based Decentralized Identity Management Scheme for Large Scale Internet of Things

Ruoting Xiong, Wei Ren, Xiaohan Hao and Yi Ren

June 27, 2021

BDIM: a Blockchain-based Decentralized Identity Management Scheme for Large Scale Internet of Things

Ruoting Xiong
School of Computer Science
China University of Geosciences
Wuhan, China

Wei Ren*
School of Computer Science
China University of Geosciences
Wuhan, China
Guizhou Provincial Key Laboratory of Public Big Data
Guizhou University
Guiyang, China
Key Laboratory of Network Assessment Technology
CAS (Institute of Information Engineering, Chinese Academy of Sciences)
Beijing, China
weirencs@cug.edu.cn

Xiaohan Hao
School of Computer Science
China University of Geosciences
Wuhan, China

Yi Ren
School of Computing Science
University of East Anglia
Norwich, UK

***Abstract*—In large scale Internet of Things, centralized authentication imposes many new challenges such as vast identity management, authentication overhead, and single point of failure. The distributed identity management is envisioned as a promising method. However, performance evaluation is still not extensive. In this paper, we proposed a decentralized identity management scheme based on blockchain, which can be applied for large scale Internet of Things such as VANET. We develop smart contracts to support large scale user access and design penalty mechanism which can prevent various types of attacks in distributed identification. The experiment results justified that our scheme can be employed for the distributed management of a large number vehicles in insurance scenarios. Specifically, the system response is 'ms' level and all the functions consumes within 250ms. Also, the time consumption of query is stable with the increase of users between 80ms to 130ms.**

***Index Terms*—Blockchain, loT, Distributed Identification, Verifiable Credential**

I. Introduction

In centralized identity management, a trusted central server or third-party has to be employed to store identity information to providing identification service upon request. As a result, traditional methods may impose two constraints: The central authority needs to maintain an increasingly large database and then the response delay may prolong accordingly. In addition, central database management imposes growing risks in that the information leakage may lead to data loss of all users. Besides, although user authorization sometimes can be provided by exchanges between platforms, they still have to register their accounts and passwords for many times. As a result, decentralized identification schemes have been attracted remarkable attentions. In [1], the registration schemes consisting of Distributed Identifier (DID) and Verifiable Credentials (VC) can be implemented in information centric network (ICN) and edge ICN scenarios, and the scheme is compatible with World Wide Web Consortium (W3C) DID standard. However, the scalability of DID-based method has not been extensively evaluated. The number of routers in ICN is small and the number of DID access is only 200.

In Internet of Things (IoT), the identity management confronts similar difficulties or even more challenges due to the special characteristics of IoT nodes. The advantages of distributed identity management are that users can generate their own key, carry it and verify their own identity. It is promising to apply distributed identity management methods in IoT to eliminate global database and enable large-scale node identification. For example, in VANET, tens of thousands of cars are in a mobile wireless networks and the global centralized database may not be able to handle the rapid growth of mobile users and provide fast authenticated access with low delay.

We observe that, blockchain can also provide new approaches for decentralized identity management (DIM) [2]. However, given the information is stored on the blockchain, the delay of accessing the data is a new challenge. Therefore, to shorten the search delay of personal information in the linked list on blockchain should be addressed firstly. In 2019,

Sara Benouar et al. [3] introduces a new architecture called Security and Storing Manager (SSM) to enable unlimited storage capability in blockchain. The proposed SSM achieves the security goals and restricts the user access In loT-Blockchain. In addition, the low latency of smart contract is also implemented, which is considered a challenge in many IoT devices applications.

There are few exploration on the performance of DID-based method in IoT. In this paper, we propose a blockchain based DIM scheme for large scale IoT. In our proposed scheme, trusted third party can be averted, and instead, peer-to-peer authentication transactions can be conducted. Also, we design a smart contract based mechanism for penalty and reward. The prototype system can be applied in several scenes such as vehicle and owner registration, insurance compensation, and incident response. The contribution of this paper is as follows:

- We propose a blockchain-based distributed identification management scheme for large scale IoT, especially, for trust management on vehicles for vehicular insurance without a trusted third party.
- We propose a smart contract based reputation evaluation method in terms of penalty and reward for defending against malicious behaviors.
- Our scheme is lightweight and is extensively evaluated by deploying only three smart contracts over off-the-shelf Etherum.

The rest of the paper is organized as follows: Section III introduces the background. In Section II, related work is reviewed. Section IV describes the proposed scheme, followed by Section V performance and safety analysis. We conclude the paper in Section VI.

II. Related work

A. Distributed Identification

There are several security problem in proposed centralized identification schemes, such as data loss and inter-platform authorization trouble. Therefore, scholars have designed and proposed many distributed identification schemes. For example [4], this paper proposes the concept of Self-Sovereign Identity (SSI) and uses OpenID Connect (OIDC) as middleware to provide distributed ledger services. In [5], a user-centered scheme is proposed. The solution to implement Verifiable Credentials (VC) in Fido's UAF framework has been mature and used in bank scenarios, which greatly solves the two problems of long account opening time and fraud problem. In [6], there is a dataset agent to store the private key and the issued VC, which is sent by the University agent. Data set agent (DSA) is a cloud based software agent, in which the data is presented by DSA and public DID. In 2017, Peralta et al. puts forward the distributed progressive fingerprint authentication [7], it classifies the fingerprint information based on feature fusion and selection, and then stores it in a distributed database to realize distributed authentication. However, it is just distributed storage data, not really distributed identification. Also, we can see in [8], the average consensus method is used to find out the most critical nodes in distributed authentication. However, they only pay attention to the algorithm level of distributed authentication, but not its application. Blockcert [9] is developed on blockchain-based SSI by MIT lab. This system allows hash of credential stored on chain. However, there is no user registration constraint and open standards for identities (such as DID). In fact, there are other systems that use blockchain but without adopting SSI for identity management. VECefblock [10] is a student credential management system in Vietnam, in which the unique national ID represents the students' identities and R. Mishra [11] uses Ethereum smart contracts to manage identities. The system stores hashed credentials on Ethereum. We compare our scheme with other blockchain-based identity management systems, along some key properties: identity management scheme (IDM), on-chain data, large-scale access (LSA), and time constrained access (TCA) by verifier. The comparisons are shown in Table I.

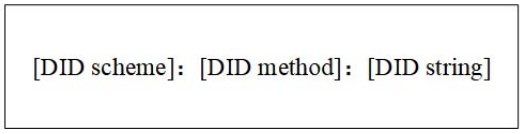

Fig. 1. The form of DID.

III. Preliminaries

In this section, we will introduce the preliminaries of our scheme, including some definitions and related work. In November 2019, a distributed identifier working group called W3C released the first public working draft of the distributed identification specification [12].

A. Distributed Identifier(DID)

DID is a verifiable and decentralized identifier of digital identity, which has string form like "did : example : 12345678abcdefg". It has three fields: DID scheme, DID method and DID string. DID scheme is a fixed field like "did". DID method defines how to generate DID, parse and verify, modify and revoke DID documents, which can be customized and registered in W3C website. For example, the school's DID is "did : edu : ***", hospital's DID is "did : hospital : ***", and user's DID is "did : user : ***". DID string is the unique identification string based on DID method, which can be formed of the ID number of users or the address of blockchain, thus the entire DID identifier is globally unique. The form of DID is shown in Fig. 1.

B. DID Document

DID document is the parsing of DID and the parsing is done by servers or database. It describes how to use a specific DID. The DID document includes the unique DID identifier, the public key, the detailed information of the public key, and other attribute descriptions of the DID holder. A DID corresponds to a DID document, which refers to the objects described in the DID document, the user's public key is held by the user

TABLE I
THE COMPARISON OF DIFFERENT IDM SCHEME

Scheme	IDM	on-chain data	LSA	TCA
Our scheme	Smart contract for IDM	DID, hash of credential, record table	Yes	Yes
Sara [3]	Smart contract for IDM	SSM, hash of data	Yes	Yes
Blockcerts [9]	SSI	Hash of credential	Not considered	Yes
VECefblock [10]	Unique national ID	Credential	Not considered	Not considered
Zoltan [2]	SSI OIDC provider	DID, VC-based PKI	Yes	Not considered
R. Mishra [11]	Smart contract for IDM	Hash of credential	Not considered	Not considered

who owns the DID, and the timestamp is the creation time and change time of the DID document. Besides, the public key information cannot be changed without authorization and only the holder of DID document can change the information of its DID document.

C. Verifiable Credentials (VC)

Verifiable certificate is equivalent to the digital certificate issued by authority CA (Certification Authority) in PKI (Public Key Infrastructure) system. The user sends some descriptive statements to the issuer, and the issuer signs the user's attributes to prove the authenticity of these attributes. The metadata includes the DID of the signer, the signature time and the signature type. Besides, claims are the attribute information that users fill in when they apply, including name, e-mail address, age, occupation, etc. The proof includes the value of the metadata and declaration signed by the issuer, and the public key used by the signature to ensure that VC can be proved.

D. Verifiable Presentation (VP)

VP is a verifiable presentation which a holder can present himself to the verifier. VP represents the signature of VC content with time and challenge random number of users. The time and challenge number presented in VP can avoid reply attack.

E. Blockchain

In 2008, an electronic currency named bitcoin was first proposed by Nakamoto [13]. Blockchain is originated from bitcoin and the nature of blockchain is a distributed shared ledger and database. It has many characteristics, such as decentralized, tamper proof, traceable and transparent. Besides, smart contracts can be developed to realize any decentralized application on blockchain [14], [15].

The data structure of blockchain is a chain. The block contains the head and the block body. When the nodes of the blockchain operate on the chain, they need to generate a transaction, which can only be added at the end of the chain. Nodes need to broadcast the request to the whole network, and other nodes use the consensus algorithm to jointly verify. If the verification is passed, the transaction will be recorded. Otherwise, they refuse to write it into the account book. Besides, smart contracts can be developed on blockchain. It is worth that the content of smart contracts cannot be tampered with, and each user who calls smart contracts must abide by the content of the contracts.

IV. PROPOSED SCHEME

A. System Overview

There are three roles in our distributed identification system: User, Issuer and Verifier, and all of these roles need to register DID. As mentioned before, the DID standard defines four parts: DID, DID document, VC and VP. Through registration, DID and DID document can be automatically linked and they will both be stored on blockchain. After registering the DID, the user can apply for VC from the issuer. If the issuer judges that the user is legal, he will issue VC to the user. When users desire to make a transaction with verifiers, they will use challenge - response method. That is, after users receive the challenge number sent by verifiers, he will sign the VC and challenge number together to create a VP. Next, users with VC can show the VP to the verifier for pre-transaction verification. Verifiers then verify the existence of DID and the validity of VP. If the verification is successful, the transaction will be carried out. The identification process is shown in Fig. 2.

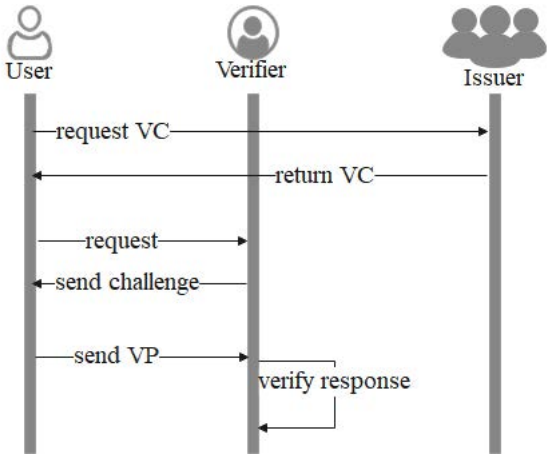

Fig. 2. System Overview: User, Verifier and Issuer.

B. DID CRUD Design

"DID CRUD" represents 'Create', 'Read', 'Update' and 'Delete' function of DID. We use the user's blockchain address as the DID string. In blockchain, the user's key is self generated and self managed. P_k means the user's public key and S_k means the user's private key. First, the user generates the blockchain address based on a pair of keys, and then calls smart contract to register DID. That is, the user provides public key and blockchain address to create the

DID and DID document. It is worth to note that issuers need to be linked in advance, and new blocks generated by later users are connected at the back of the chain. The difference between users and issuers is that the method fields of DID are different. For example, the DID's method with "university" and "company" indicates the issuer. After the data is packed upload on blockchain, we claim that both of DID and DID documents are true and reliable.

With the rapid growth of access users, the total amount of DID will grow rapidly and the scale will be huge. In order to quickly parse the DID document, our scheme use smart contract to maintain a record table to store the latest update block height of the DID document, instead of maintaining a global database to record DID and DID document. The content is: DID and the latest record block height of DID document. That is, as long as the block height is found, the latest DID document can be parsed to get the P_k. The process of uploading DID on blockchain is shown in Fig. 3.

In addition, we propose the method of changing the DID document and revoking their identity for users. After passing authentication, the user can modify DID document. The method of revoking distributed identity is deleting the records in the DID table. Next, the specific method of challenge - response is using the public key issued by the user in the DID document to encrypt the random number and send it to the DID document holder. Only the real private key holder can decrypt the random number, and then sign the random number with the private key and send it to the verifier. The verifier uses the public key to verify whether the obtained number is equal to the random number, which indicates that the public key information in the document is correct and the user is a real private key holder.

C. VC and VP Design

Verifiers verify the user's VP before the transaction, and the transaction can only be performed after the verification is passed. Firstly, the user requests a transaction from the verifier, and then the verifier sends the challenge number to the user. Secondly, the user connects the VC with the hash value of the random number, signs it with his own private key to generate a VP and then sends it to the verifier. After getting the signature, verifier calls the search function of DID smart contract to obtain the user's public key and verify the signature with the public key. The intermediate result is the hash value of VC and challenge number. Verifier hashes its own challenge number and compares it with the sent value. If the result is equal, it can further verify the authenticity of VC. The method of verifying VC is to check whether it exists on chain. In a word, if the VP verification is successful, all of the following conditions need to be met:

(1) VP is signed by the user.

(2) VC is signed by issuer.

(3) Users and issuers have DIDs on the chain.

(4) The correctness of time or random number.

D. IoT Application Design

The traditional insurance claims procedure is to inform the insurance company of the car accident. And insurance company will determine the loss of users who bought the car insurance. However, there are some problems with the traditional insurance claims process. For users, the first is the management of paper documents and second is to report the accident on time within 48 hours. For companies, the insurance companies need to handle huge record of users and since the results of loss determination and claims may be tampered, there is a lack of means to verify their reality.

In the scene of vehicle and owner registration, and insurance payment credible transfer, our participants include the car owner, car shop, driving school, and insurance company. Among the participants in our scheme, issuers are car shops, driving schools and insurance companies, users are car owners, and verifiers are insurance companies. They all need to register the DID to get access to the system and their forms of DID are listed in Table II:

TABLE II
THE FORM OF DID

Entity	Role in DID	The form of DID
Insurance company	Issuer & Verifier	"did : company : ca35b...3c"
Car onwer	User	"did : cuser : 4b08...d2db"
Car shop	Issuer	"did : store : ..."
Driving School	Issuer	"did : dschool : ..."

Specifically, the paper documents that need to be prepared before the vehicle owner can drive the vehicles on the road are: (1) The driving license of the motor vehicle. (2) The driver's license of the vehicle owner. (3) The insurance sheets of the insured vehicle. After the car owner has an accident, the insurance company determines the loss for the car and users can obtain compensation.

The replacement relationship between DID and VC and the above files is shown in the following Table III:

TABLE III
PAPER DOCUMENTS VS VCS

Paper document	VC
Vehicle license	Vehicle VC
Driver license	Driver VC
Vehicle insurance	Insurance VC
Vehicle compensation	Compensation VC
ID card	DID and DID document

The flow of the scheme is listed below, and the design of four VCs are shown in Fig. 5:

- **DID.** The car owner, car shop, driving school, insurance company register DID respectively.
- **Vehicle VC.** After the owner buys a car, the car store generates the vehicle VC for the owner. The claims include: 'car number', 'car type', 'car color' and 'driver DID', etc.
- **Driver VC.** The driver passes the driving license examination, and the driving school generates the driver VC

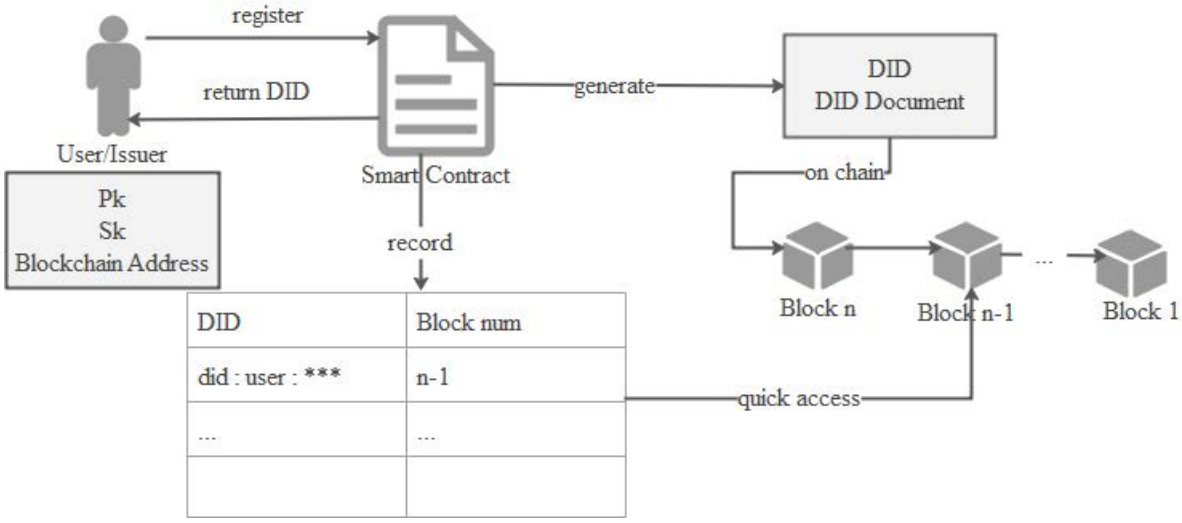

Fig. 3. The process of uploading DID on blockchain.

for the driver. The claims include: 'car type' (the types of cars can be allowed to drive), 'license number', 'license due time' and 'driver DID', etc.

- **Insurance VC.** When the car owner purchases insurance, the insurance company generates an insurance VC for the car owner after the vehicle VC verification. The claim includes: 'insurance type', 'insurance number', 'car number' and 'driver DID'.
- **Compensation VC.** When determining the loss of vehicles, the Internet of vehicles can restore the data before the accident. In the event of an accident, according to the monitoring information in the vehicle and the monitoring situation of the vehicle surface and road, the loss situation is automatically generated, so as to carry out the loss assessment. The user will send the damage claims to the insurance company, requesting the insurance company to issue compensation VC. The insurance company should verify the insurance VC before issuing compensation VC. The claims include: 'compensation money', 'compensation number', 'car number', 'accident results' and 'driver DID'. The process is shown in Fig. 4.

E. Penalty and Reputation Mechanism Design

After verifying the compensation VC, the insurance company will pay the loss. Some malicious users will try to verify fake VC and attempt to forge the issuer signature of VC. Therefore we develop penalty mechanism to reduce users' security deposit when the verification fails. Also, we develop the reputation mechanism to record the number of insurance payments made by users.

Penalty mechanism. Any user (holder) who calls the verified VP smart contract needs to deduct the security deposit. During the verification process, there may be situations of success and failure.

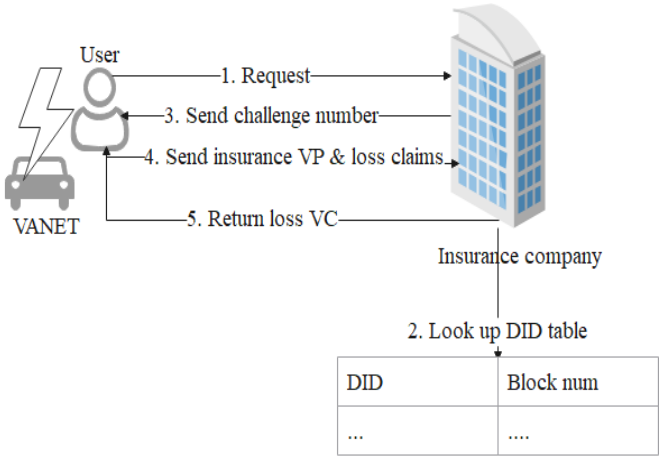

Fig. 4. The process of issuing compensation VC.

If the verification fails, it can be the following situations:

- Users not found. Therefore, the user's public key cannot be obtained from blockchain to verify the signature, so no transaction is performed.
- VC not found. After obtaining the user's public key, verifiers will verify the existence of VC on the chain. If VC does not exist, the security deposit that the user gives to the verifier is deducted. If VC exists, but the challenge number is not correct, it indicates that the user is a fake attacker, then the verification fails and no transaction is performed.

If the verification is successful, the following conditions have to be met:

- The user's signature is verified successfully.
- The sent VC exists on blockchain and it must be issued by issuers.

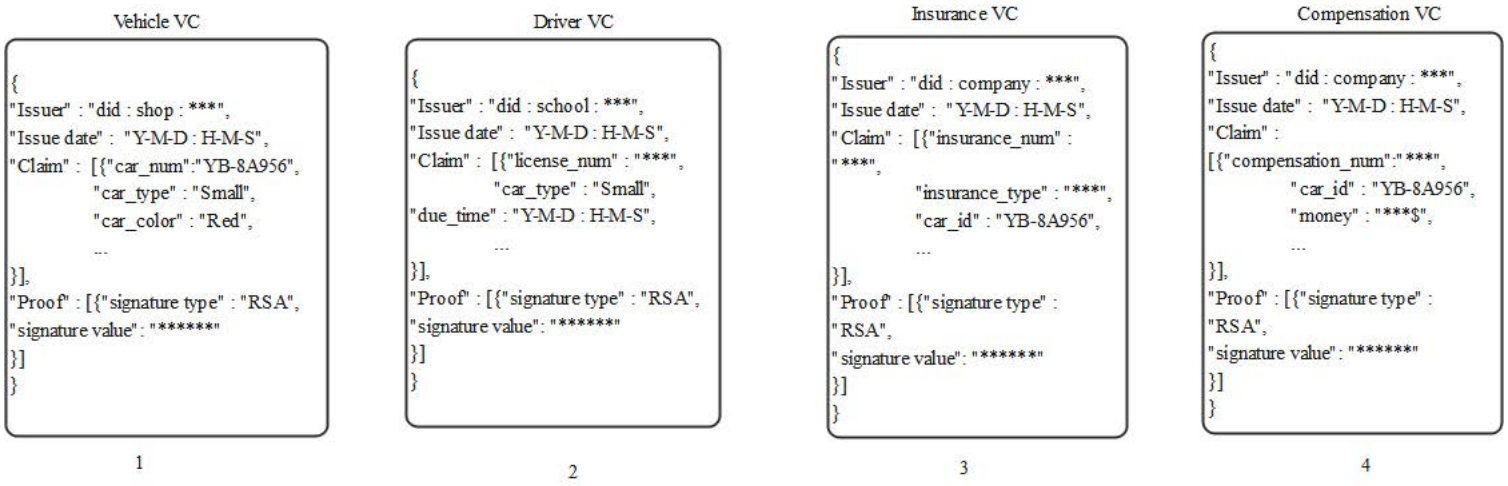

Fig. 5. Example of VC.

- The number of challenges is equal to the number sent.

Reputation mechanism. The reputation mechanism states that the more times a user verifies VP, the less credit he has. After each verification of VP passes, the smart contract will add the times of compensation in the table record. Insurance company can make it clear through inquiring the total number of times a user has verified VP and get the insurance pay. Typically, for a user who verifies VP for more than three times a year, the insurance company will consider no longer allowing the user to purchase insurance, that is, no longer issuing an insurance VC to the user. The algorithm of these two mechanism is shown in Algorithm 1.

Finally, we can get a record table of reputation mechanism. The table is shown in Table IV.

TABLE IV
RECORD OF REPUTATION MECHANISM.

DID	Times of compensation
did : user: ***	1
did : user: ***	3
...	...

Algorithm 1 Penalty and Reputation Functions.

procedure PENALTY PROCEDURE
  $challengenum \leftarrow$ random sent by verifier
  $didexist \leftarrow$ if DID exists in table
  $vcexist \leftarrow$ if VC exists on chain
  $vp \leftarrow$ times of compensation
  $Tableu \leftarrow$ table record registered DID
Start:
  Deduct verifier's security deposit.
  Look up Tableu for User's publickey.
  if *verifysignature*$(User)$ **then**
    Look up VC on chain.
    if VC exists on chain **then**
      if ! $challengenum$ is repeat **then**
        Pass the verification.
        Return the verifier's security deposit.
        goto *Reputation.*
      end if
      Deduct user's security deposit for
      false challengenum.
    end if
    Deduct user's security deposit for false VC.
  end if
  Deduct user's security deposit for false VP.
Reputation:
  if ! $didexist$ **then**
    insert(DID,1).
  end if
  update(DID, *vp*+1).
end procedure

V. EXPERIMENTS AND ANALYSIS

Hardware and Software. In our experiments, we use Ganache to develop blockchain application. We use Web3 module to write Python program to interact with nodes in Ganache blockchain, that is, Web3 is the interface between users and blockchain nodes. Truffle is used to write smart contracts, which can be deployed and invoked on Ganache. The system environment configuration is shown in Table V.

A. System Performance

Gas consumption. We count gas consumption for DID smart contract and VC smart contract and VP smart contract.

TABLE V
SYSTEM ENVIRONMENT CONFIGURATION.

Project	Hardware version or condition
Operating system	Windows 10
Environment	Python3.7, Truffle, Ganache, Web3
Programming Language	Solidity

As mentioned above, DID smart contract can achieve DID's CRUD, and VC smart contract can publish VC for users. VP smart contract will provide VP verification function and have penalty and reputation mechanism. As shown in Fig 6, the gas consumption of 'read' DID is relatively low compared to other functions, which means users can easily get information on the blockchain, and the gas consumption is small. Besides, the gas consumption of 'verify' DID is small, indicating that users can also identify themselves with little gas loss.

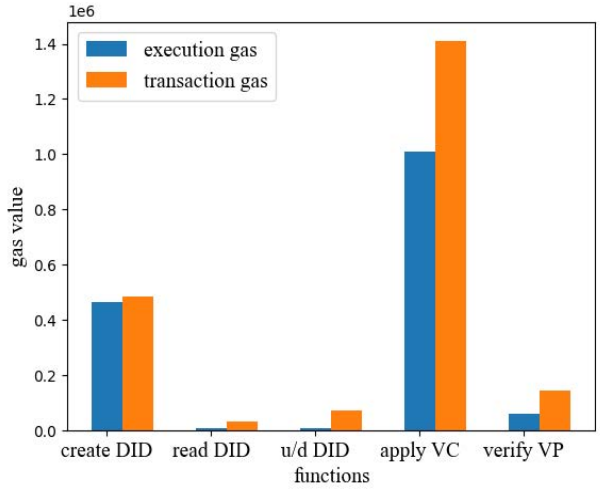

Fig. 6. The gas value of smart contracts.

Time consumption. We also evaluate the time consumption of every function and the results are shown in Fig 7. To conclude, all functions are implemented at the 'ms' level, which means the users can get quick response when using our system. The time of 'create DID' is about 200ms, and the time of 'u/d DID' is over 200ms. In addition, the time of 'read DID' and 'verify VP' are less than 150ms. The results are all reasonable since the functions of 'create' and 'apply' will lead to the creation of new blocks, which costs more time. And the time of 'u/d DID' include not only the time of new block creation, but also the time of changing the table records.

Besides, we compare the average time consumption of user query function among different schemes and the results are listed in Table VI.

B. Security Analysis

Avoid Replay Attack: The attacker can replay the account and password used by the user by recording past communication information, and pass the server verification. This attack

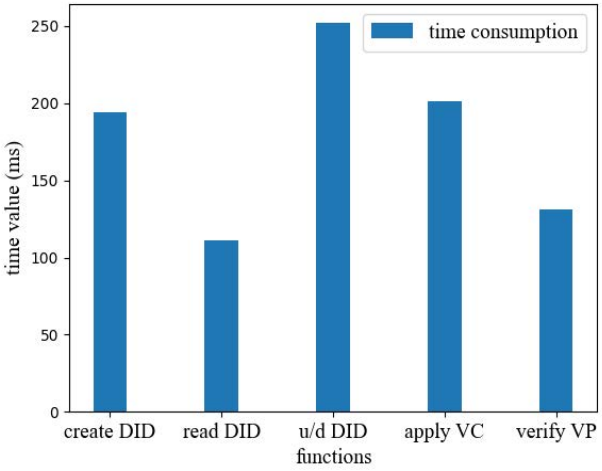

Fig. 7. The time consumption of every function.

TABLE VI
THE COMPARISON OF AVERAGE TIME CONSUMPTION.

Scheme	Query time
Our Scheme	120 ms
Sara [3]	500 ms
VECefblock [9]	150ms

can be resisted by challenge - response. In our scheme, before the users send VC to verifiers, they will add information about time and challenge random numbers to generate VP. Through time and different random numbers, legitimate devices can detect malicious replay attacks.

Prevent User Impersonation Attack: The attackers impersonate legitimate user and pretends to be the key holder. In our scheme, the attackers can not get the user's private key, and we can verify the authenticity of the user's identity by zero knowledge proof.

Central Server Attack: The attackers deliberately attack the central database, resulting in the loss of all personal information. Since our distributed identification system is implemented on blockchain, it will not have a central server. All personal information is on distributed ledger and everyone can have access to it, thus our scheme can resist central sever attacks.

C. Scalability Analysis

We show the time consumption of 'read DID' does not change rapidly with the growth of users. The results are show in Fig 8. The time is changing between 80ms and 130ms, which shows that the system is stable when handling large access of users. Compared with scheme 1 and 2, the low latency of large-scale access shows the scalability of the scheme, which is essential to loT-Blockchain.

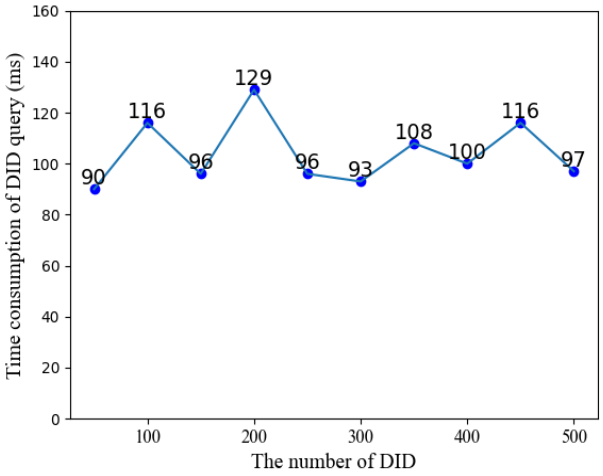

Fig. 8. The time delay of DID query with the increase of users.

VI. Conclusion

We proposed a decentralized identity management scheme and implemented a prototype over off-the-shelf Etherum. The scheme can be applied in large scale IoT as distributed blockchain can be accessed by large scale IoT nodes. The smart contract based mechanism is also proposed for reputation management, which consists of penalty and reward counting system to prevent potential attacks, such as reply attacks, user impersonation attacks and central server attacks. After deployed and evaluated the smart contracts, we also verified that the gas consumption of user DID query and VP verification is relatively low. The detailed analysis also showed that the scheme is feasible and secure, since the time consumption of query is stable regardless of the increase of users and all the users' request will be responded in 'ms" level.

Acknowledgment

The research was financially supported by National Natural Science Foundation of China (No. 61972366), the Provincial Key Research and Development Program of Hubei (No. 2020BAB105), the Foundation of Key Laboratory of Network Assessment Technology, Chinese Academy of Sciences (No. KFKT2019-003), Major Scientific and Technological Special Project of Guizhou Province (No. 20183001), and the Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2018BDKFJJ009, No. 2019BDKFJJ003, No. 2019BDKFJJ011).

References

[1] Alzahrani, B.A., 2020. An information-centric networking based registry for decentralized identifiers and verifiable credentials. IEEE Access PP, 1–1.

[2] Lux, Z., Beierle, F., Zickau, S., Göndör, S., 2019. Full-text search for verifiable credential metadata on distributed ledgers.

[3] S. Benouar and A. Benslimane, "Robust Blockchain for IoT Security," 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1-6, doi: 10.1109/GLOBECOM38437.2019.9013580.

[4] Lux, Z.A., Thatmann, D., Zickau, S., Beierle, F., 2020. Distributed ledger- based authentication with decentralized identifiers and verifiable credentials. arXiv:2006.04754.

[5] Laborde, R., Oglaza, A., Wazan, S., Barrere, F., Venant, R., 2020. A user-centric identity management framework based on the w3c verifiable credentials and the fido universal authentication framework, in: 2020 IEEE 17th Annual Consumer Communications and Networking Conference (CCNC).

[6] Barclay, I., Radha, S., Preece, A., Taylor, I., Nabrzyski, J., 2020. Certifying provenance of scientific datasets with self-sovereign identity and verifiable credentials. arXiv:2004.02796.

[7] H. Liu et al., "Distributed Identification of the Most Critical Node for Average Consensus," in IEEE Transactions on Signal Processing, vol. 63, no. 16, pp. 4315-4328, Aug.15, 2015, doi: 10.1109/TSP.2015.2441039.

[8] Peralta D, Triguero I, García S, et al. Distributed incremental fingerprint identification with reduced database penetration rate using a hierarchical classification based on feature fusion and selection[J]. Knowledge-Based Systems, 2017, 126: 91-103.

[9] Blockcerts:the open standard for blockchain credentials, 2020. https://www.blockcerts.org/, Accessed: 07-05-2020.

[10] Nguyen BM, Dao T, Do B. 2020. Towards a blockchain-based certificate authentication system in Vietnam. PeerJ Computer Science 6:e266 https://doi.org/10.7717/peerj-cs.266

[11] Raaj Anand Mishra, Anshuman Kalla, Nimer Amol Singh, Madhusanka Liyanage: Implementation and Analysis of Blockchain Based DApp for Secure Sharing of Students' Credentials. CCNC 2020: 1-2

[12] D. Longley, M.S., 2021. Decentralized identifiers (dids) v1.0. URL: https://www.w3.org/TR/did-core/.

[13] Pierro, M.D., 2017. What is the blockchain? Computing in Science and Engineering 19, 92–95.

[14] Wang, B., Zhao, S., Li, Y., Wu, C., Yukita, K., 2021. Design of a privacy-preserving decentralized energy trading scheme in blockchain network environment. International Journal of Electrical Power and Energy Systems 125, 106465.

[15] A, N.S., B, L.T.A., D,W.L.C., E, X.Q., E, K.Y., 2020. A blockchain empowered aaa scheme in the large-scale hetnet - sciencedirect. Digital Communications and Networks.