


Research Article

Trusted Blockchain-Based Signcryption Protocol and Data Management for Authentication and Authorization in VANETs

Jinqi Su,¹ Runtao Ren ,² Yinghao Li,² Raymond Y. K. Lau,³ and Yikuan Shi⁴

¹School of Economics and Management, Xi'an University of Posts and Telecommunications, Xi'an 710061, China

²School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

³Department of Information Systems, City University of Hong Kong, Hong Kong

⁴AVIC Jonhon Optron Technology Co., Ltd., Luoyang 471000, China

Correspondence should be addressed to Runtao Ren; 760473028@qq.com

Received 6 April 2022; Accepted 30 April 2022; Published 21 May 2022

Academic Editor: Maode Ma

Copyright © 2022 Jinqi Su et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Vehicular Ad hoc Networks (VANETs) are the industrial cornerstone of intelligent transportation system (ITS), which are widely used in traffic management, automatic driving, and road optimization. With the expansion of the scale of the mobile ad hoc networks (MANETs) and smart vehicles (SV), VANETs will produce a large amount of data. In the open access environment of VANETs, the security of information transmission and the authenticity of user identity need to be considered when different vehicles communicate. In order to solve the cybersecurity risks of large-scale deployment of VANET, this paper proposes a trusted blockchain-based signcryption protocol and data management (TB-SCDM) for authentication and authorization (A&A) in VANETs. In the existing attack model, TB-SCDM can ensure the confidentiality and undeniability of information, as well as can effectively resist 51% attacks, eclipse attacks and double-spending attacks, etc. Through benchmark analysis, this scheme has higher computing efficiency and lower storage cost compared with other existing schemes.

1. Introduction

The VANETs have stimulated interest in both academic and industry, thanks to their intelligence and networking that assist vehicle driving and promote the application and development of ITS (e.g., automatic driving) [1–3]. At the same time, the VANETs have also become one of the most promising and fastest-growing subsets of the MANETs [4]. The VANETs are distributed and self-organized networks which communicate through wireless media, built up by SV, roadside units (RSUs), global positioning system (GPS), trusted authority (TA), and on-board units (OBUs). SV could communicate with each other as well as with roadside units (RSU) (e.g., electric toll collection of highways), which provide a good dedicated short-range communication (DSRC) by IEEE 802.11p standard for automatic driving technology to identify real-time traffic conditions [5–7]. TA is a third-party certification center used by the RSU and OBU that is responsible for controlling the whole network. RSU is a base station (e.g., Wi-Fi

or WiMAX) that keeps as a central hub between the TA and the OBU and performs different authentications. The OBU is introduced on the vehicle to acquire procedure and exchange data identified with different vehicles and RSUs through DSRC.

With the main goal of improving road safety and driving conditions, VANETs are established with five types of communications: the vehicle-to-vehicle (V2V), vehicle-to-roads (V2R), vehicle-to-infrastructure (V2I), roads-to-roads (R2R), and the roads-to-infrastructure (R2I) [8]. The architecture of VANETs is appeared in Figure 1. Due to the open nature of VANETs and lacking infrastructure, these delays establish reliable end-to-end communication paths and have efficient data transfer [9–10]. In particular, automatic driving technology has many system problems and security difficulties in obtaining availability, securing communication, and accessibility of exchange. In VANETs, SV are strangers who do not trust each other [11]. Without authentication and authorization, the attacker may impersonate any vehicle to broadcast forged messages to easily track the target vehicle by analyzing

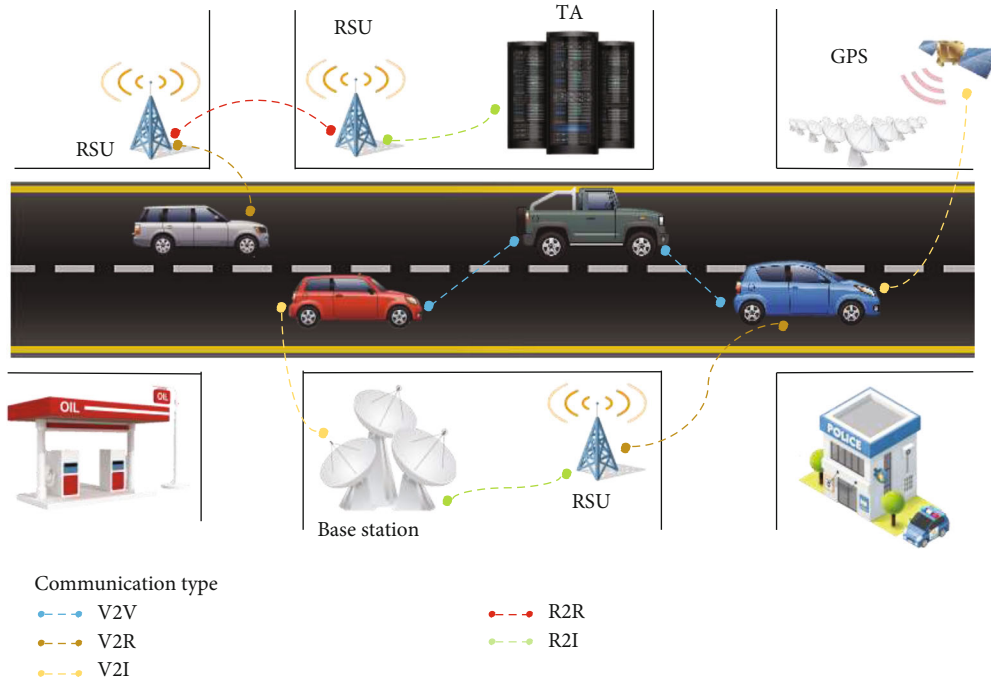


FIGURE 1: The architecture of VANETs.

the broadcast messages, which will pose a serious threat [12]. Therefore, when the users of SV use automatic driving, they need to authenticate and authorize the identity of vehicle in VANETs.

In the conventional A&A schemes, public key infrastructure- (PKI-) based solutions need a certificate authority (CA), while identification-based solutions require a key generation center (KGC) to provide vehicles with secure authentication [13–15]. However, there is a high computational overhead and large storage capacity on the CA and TA in the case of large number of certificates.

Considering the above limitations, blockchain has the function of distributed storage, which can effectively realize decentralization [16–17]. The methods of automatically injecting trust, checking reliability, monitoring interentity communication, and analyzing behavior can be implemented in the blockchain. It forms a distributed database by using digital signature, encryption technology, hash function, and timestamp [18]. Blockchain assigns the responsibility of maintaining privacy and security to all entities in VANET instead of centralized operation [19–22]. In addition, identity-based signcryption protocol has shorter ciphertext and less computational overhead, which can sign and encrypt the data to ensure the confidentiality and nonrepudiation of the information [23–25].

Our contribution: in automatic driving, since VANETs consist of a large number of SV at high speed, the security of information transmission must be satisfied A&A efficiently. In this paper, we propose a scheme that combined blockchain and signcryption to realize the A&A when the SV using automatic driving interacts with other media. Figure 2 shows the physical process of TB-SCDM when the SV use automatic driving. The contributions of this article are as follows.

- (1) This article is an SV management system built on the consortium chain, which can upload the relevant data of SV to the blockchain to realize distributed storage
- (2) The A&A function of SV users in ITS can be effectively realized in TB-SCDM scheme. The A&A mechanism we designed can ensure the trusted identity and effective authorization of SV users in VANETs
- (3) The TB-SCDM scheme combines blockchain and signcryption. The data on the consortium chain cannot be tampered with arbitrarily. This mechanism provides a stronger security level for signing and encrypting the data that needs to be verified. Therefore, the confidentiality and unforgeability of SV information transmission in VANETs can be realized through TB-SCDM

2. Preliminaries

2.1. Consortium Chains. Consortium chain has the advantages of weak concentration, high controllability, and great scalability [26]. Thanks to the number of nodes and organizational structure being relatively limited, consortium chain is mainly applied in systems built by specific organizations (e.g., data interaction of ITS). The rights of each participating node in the consortium chain are completely equal, and they can realize the trusted exchange of data. Each node of the consortium chain has a corresponding entity that wants to join and exit only to be executed after authorization. In the consortium chain, data transactions do not need the consensus of the

whole blockchain network. Therefore, the consortium chain satisfies the data management requirements of VANETs through controlled access, efficient storage and trusted storage.

2.2. Smart Contract. Smart contract refers to a computer program that can be executed by a network of mutually untrusted nodes without any trusted authority. Compared with traditional programming source code, smart contracts utilize blockchain immutable distributed storage. In the initial stage of building the data storage system of SV, the vehicle management system can write triggers to realize the functions according to the actual needs. Once the system is put into operation, when the trigger conditions are met, the content of the smart contract can be executed to complete data upload, network access, and other processing functions. Finally, smart contracts can be developed to achieve smaller permission control granularity.

2.3. Practical Byzantine Fault Tolerance. The practical byzantine fault tolerance (PBFT) means a kind of fault tolerance of distributed network (i.e., the network can still make honest nodes reach a consensus. The PBFT mechanism will specify that one node in the system is the master node, and the other nodes are secondary nodes [27]. The process of PBFT is shown in Figure 3. When the primary node fails, all legal nodes in the system are eligible to upgrade from the secondary node to the primary node and follow the principle of the minority obeying the majority to ensure that honest nodes can reach a consensus. However, in order for the PBFT to operate normally, the number of malicious nodes must be less than 1/3 of the total number of nodes in the network. For example, in order to ensure the normal operation of the whole system, assuming that the number of invalid or malicious nodes tolerated by PBFT is F and the total number of nodes of the system is $|R| = 3F + 1$, then $2F + 1$ normal nodes are required. Hence, the PBFT algorithm can tolerate less than 1/3 invalid or malicious nodes.

2.4. Meaning of Symbols. The specific meaning of the symbols is contained in Table 1.

3. Formation Definition

3.1. Syntax. The algorithm definition of TB-SCDM is as follows.

Initialize (1^θ) \rightarrow *Table*: the initialize algorithm is executed by an administrator in the securable environment. Firstly, the administrator has a query for system and takes as input a security parameter θ , then return a local table named management and output 0 otherwise.

BlockUp (*Table*) $\rightarrow 1$ or 0 : the BlockUp algorithm is run by the administrator as well. For this algorithm, administrator sends each primary key ($N_i, I_i, ID_i, IK_i, IR_i, PK_i, C_i, \mu_i, \sigma_i, \sigma_{IR_i}$) to table for achieving consensus among nodes then output 1 or 0.

Signcrypt (N_i, ID_i, IK_i) $\rightarrow PK_i, SK_i$: the Keygen algorithm is performed by one user who tries to register a new account in the system. The user sends N_i, ID_i , and IK_i to the system to

generate PK_i, SK_i and σ_i . Then, $PK_i, SK_i, C_i, \delta_i, \mu_i$, and σ_i will be saved in the table for connecting blockchain.

Authentication (N_i, IK_i, SK_i) $\rightarrow 1$ or 0 : this user sends N_i^* , IK_i^* , and SK_i^* to the system to produce digest δ_i^* and δ_i^{**} for validation. There are two cases in this process.

Case 1. If $\delta_i^* = \delta_i^{**}$, the user can realize the login process to show that the user's identity information is reliable.

Case 2. If $\delta_i^* \neq \delta_i^{**}$, the authentication of this user with identity is failed and output 0.

Update (IR_i, SK_i) $\rightarrow \sigma_{IR_i}$: this algorithm is executed by the user who needs to update the resource in the system. Assume the identity of user is valid, the IR_i and SK_i can get input by this user to output signcryptUserResource σ_{IR_i} on the block.

Authorization (N_i, PK_i) $\rightarrow IR_i$ or 0 : this Authorization algorithm is to realize the authorization of users. Initially, the user should send the target account N_i^* and the corresponding public key PK_i^* to platform for verification. There are two cases in this algorithm.

Case 1. If $\delta_i^* = \delta_i^{**}$, the user can be authorized and gain the part access for userResource IR_i .

Case 2. If $\delta_i^* \neq \delta_i^{**}$, this user failed to authorization and output 0.

Conversation (N_i, PK_i) $\rightarrow 1$ or 0 : the algorithm is used to establish dialogue between different users. First, the user can send N_i and PK_i^* to platform for communication. There are two situations in this algorithm.

Case 1. If $\delta_i^* = \delta_i^{**}$, the user can be authorized and gain a conversation.

Case 2. If $\delta_i^* \neq \delta_i^{**}$, instant messaging channel cannot establish and output 0.

Transaction (*Chain*) \rightarrow *transactionHash*: this algorithm is run by administrator in order to obtain the information on the blockchain. The administrator can query the main parameters of the blockchain to get buildTime, buildType, genesisBlockHash and contractAddress, etc.

4. Concrete Scheme

There are eight parts in the TB-SCDM: Initialize, BlockUp, Signcrypt, Authentication, Update, Authorization, Conversation, and Transaction. The steps of Authentication, Update, and Authorization are described in Figure 4.

4.1. Initialize. This algorithm is to register a table named management on the blockchain so that later users' information can be registered on the consortium chain.

4.2. BlockUp. This algorithm is executed by the administrator. Its purpose is to create each primary key in the table

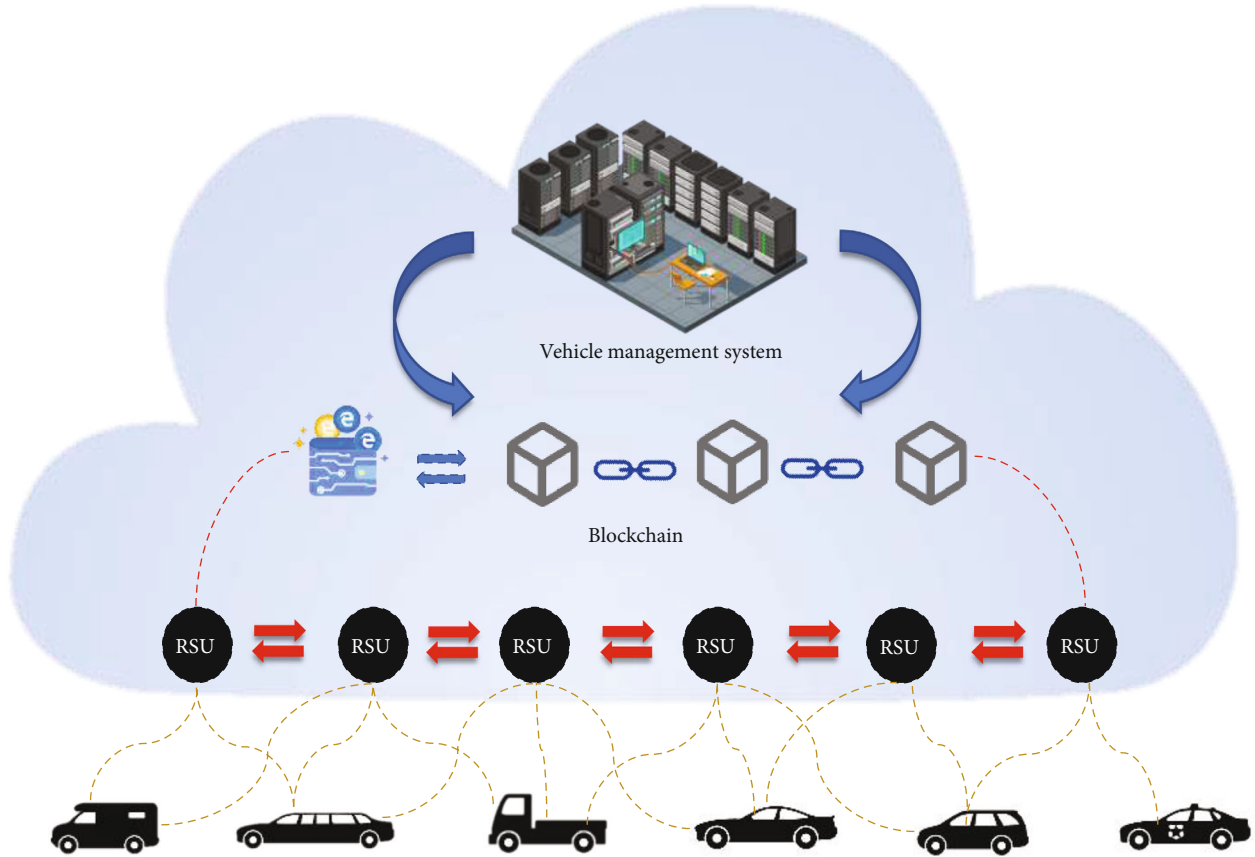


FIGURE 2: The process of TB-SCDM.

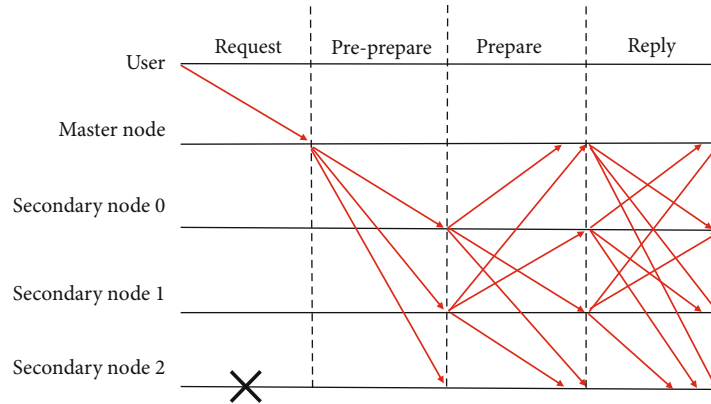


FIGURE 3: The process of PBFT.

generated in algorithm 1 and then upload the data of each primary key to the blockchain.

4.3. *Signcrypt.* Firstly, the system will first give a public-private key pair to the user with identity I_i . Accordingly, the user will deposit the I_i , ID_i , and IK_i in the plainText M_i to generate the hash value δ_i . Then, this user utilizes the private key SK_i to produce the signature μ_i and utilizes the public key PK_i to

encrypt M_i for getting the ciphertext C_i . Finally, μ_i and C_i will be merged to return the signcrypton σ_i .

4.4. *Authentication.* For Authentication algorithm, the user of identity I_i can input N_i^* , IK_i^* , and SK_i^* in the system and then query whether there exists the account named N_i^* in the table. If exist N_i^* , this user will enter the authentication stage.

TABLE 1: Specific meaning of symbols.

Notations	Meaning
N_i	The accountName of user
I_i	The identity of arbitrary user
IK_i	The userKey of I_i
ID_i	The userData of I_i
IR_i	The userResource of I_i
PK_i	The publicKey of I_i
SK_i	The secretKey of I_i
δ_i	The digest of I_i
M_i	The plainText of I_i
C_i	The cipherText of I_i
μ_i	The signature of I_i
$?_i$	The signcryption of I_i
σ_{IR_i}	The signcryptedUserResource of I_i

On the client side, the $(I_i, ID_i, \text{ and } IK_i^*)$ will deposit in plainText M_i to produce the hash value δ_i^* . Accordingly, the signature μ_i^* can be generated by the private key SK_i^* .

On the blockchain side, the signature μ_i^* can be unsigned by the trusted public key PK_i stored by previous user of identity I_i . Accordingly, the trusted signcryption σ_i can be unsigned to get the hash value δ_i^{**} . After obtaining the above data, the next step will verify the user's identity. There are two cases in this process.

Case 1. If $\delta_i^* = \delta_i^{**}$, the user can realize the login process to show that the user's identity information is reliable.

Case 2. If $\delta_i^* \neq \delta_i^{**}$, the authentication of this user with identity is failed and output 0.

4.5. Update. The user of identity I_i can update the resource in the system through this algorithm. The user can input IR_i and SK_i in the system. Then, I_i , PK_i , and IR_i will be merged into the plainText M_i . The following queries are same as those in Algorithm 1.

Finally, the updated information of these users will be uploaded to consortium chain.

4.6. Authorization. This algorithm is designed to authorize the legitimacy of user's behavior. In the authorization process, we add the token technology. In this mechanism, we first set the upper limit of the user's single query time to 300 s.

After exceeding the time, the user's access rights will disconnected, and his identity needs to be verified newly. Within legal time, account N_i^* will be first verified for existence. If account N_i^* exists, then the user of identity I_i will enter the authentication stage for authorization.

On the blockchain side, the trusted signcryption σ_i can be unsigned to return the signature μ_i^* and ciphertext C_i^{**} . Accordingly, the signature μ_i^* can be unsigned to get

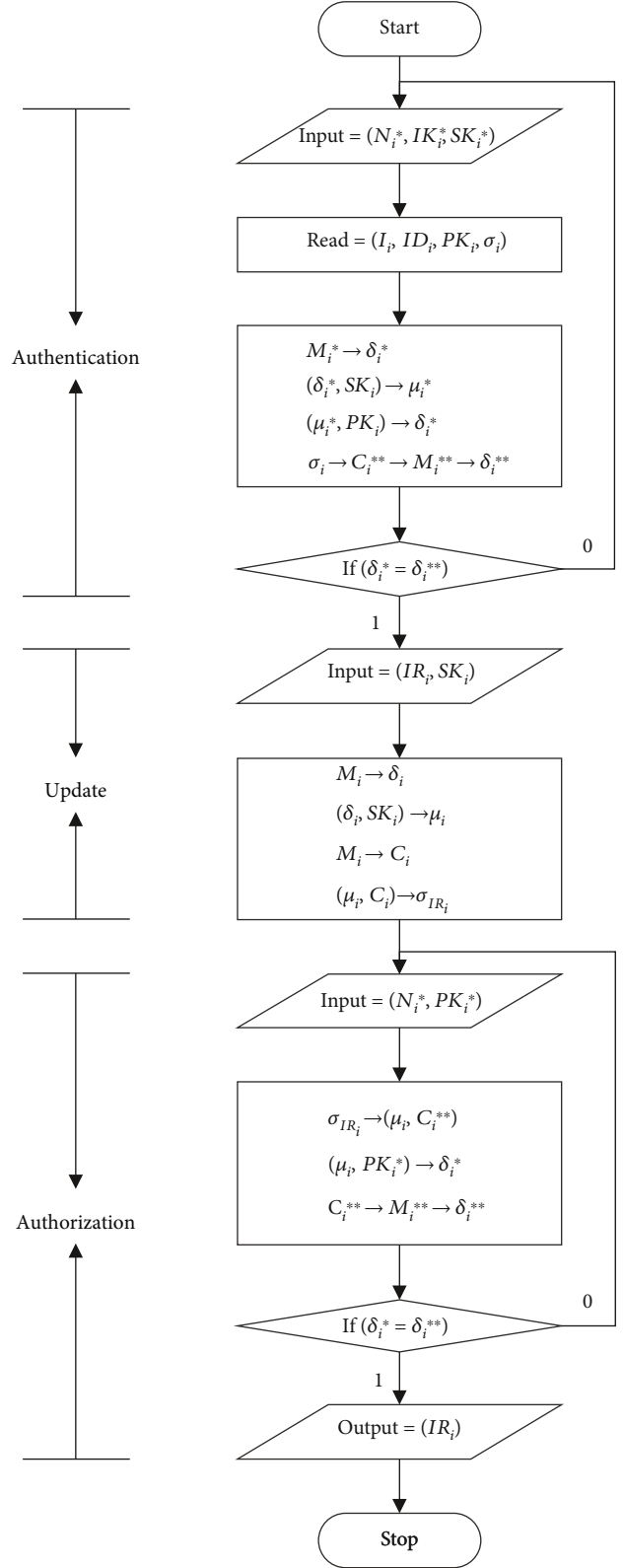


FIGURE 4: The process of Authentication, Update, and Authorization.

```

Input:  $I^{\theta}$ 
Output: Table
createTable() private
{
  tf.createTable("management", "I", "Ni", "IDi", "IKi", "PKi", "μi", "Ci", "σi", "IRi", "σIRi");
}
openTable() private returns(table)
{
  TableFactory tf = TableFactory(0x1001);
  Table table = tf.openTable("management");
  return table;
}

```

ALGORITHM 1: Initialize.

```

Input:  $I_i, N_i, ID_i, IK_i, PK_i, \mu_i, C_i, \sigma_i, IR_i, \sigma_{IR_i}$ 
Output: true or false
statu = select(management)
if (statu != 0) {
  Table table = openTable();
  Entry entry = table.newEntry();
  entry.set("I, Ni, IDi, IKi, PKi, μi, Ci, σi, IRi", Ii, Ni, IDi, IKi, PKi, μi, Ci, σi, IRi);
  return True;
} else {
  return false;
}

```

ALGORITHM 2: BlockUp.

```

Input:  $N_i, ID_i, IK_i$ 
Output:  $PK_i, SK_i$ 
Function SignCrypted Input( $M_i, SK_i$ ) Output( $\delta_i, \mu_i, C_i, \sigma_i$ ){
   $\delta_i$  = Method.hash( $M_i$ );
   $\mu_i$  = Method.sign( $\delta_i, SK_i$ );
   $C_i$  = homomorphicEncryption.Enc( $M_i$ );
   $\sigma_i$  = signcrypt( $\mu_i || C_i$ );
}
statu = select( $N_i$ );
if (statu != 0) {
  Get cryptographic KeyPair = new createKeyPair();
  Get cryptographic Method = new cryptographic (CryptoType.SCHNORRTYPE);
   $PK_i$  = KeyPair.getPKi ();
   $SK_i$  = KeyPair.getSKi ();
   $M_i$  =  $I_i || ID_i || IK_i$ ;
   $\delta_i || \mu_i || C_i || \sigma_i$  = function.SignCrypted( $M_i, SK_i$ );
  entry.set("Ni, IDi, IKi, PKi, μi, Ci, σi", Ni, IDi, IKi, PKi, μi, Ci, σi);
  count = table.insert(Ni, entry);
  if (count == 1) {
    statu_code = true;
  } else {
    statu_code = false;
  }
} else {
  statu_code = false;
}
return statu_code PKiSKi;

```

ALGORITHM 3: Signcrypt.


```

Input:  $N_i, IK_i, SK_i$ 
Output: true or false
Function unSignCrypted Input( $\sigma_i, PK_i$ ) Output(true or false)
{
  ' $\mu_i^*$ ' || ' $C_i^{**}$ ' = unencrypt( $\sigma_i$ );
  if('' $\mu_i^*$ ' == 0) {
    ' $\mu_i^*$ ' = ' $\mu_i^*$ ';
  }
  ' $\delta_i^*$ ' = Method.unsign('' $\mu_i^*$ '',  $PK_i$ );
  ' $M_i^{**}$ ' = homomorphicEncryption.Dec('' $C_i^{**}$ '');
  ' $\delta_i^{**}$ ' = Method.hash('' $M_i^{**}$ '');
  if('' $\delta_i^*$ '' == ' $\delta_i^{**}$ '') {
    statu_code = true;
  } else {
    statu_code = false;
  }
}
statu = select( $N_i$ );
if(statu != 0) {
  Get cryptographic Method = new cryptographic (CryptoType.SCHNORRTYPE);
  ' $IK_i^*$ ' = result.getValue2();
  ' $SK_i^*$ ' = result.getValue3();
  ' $M_i^*$ ' =  $I_i$  ||  $ID_i$  || ' $IK_i^*$ ';
  ' $\delta_i^*$ ' = Method.hash('' $M_i^*$ '');
  ' $\mu_i^*$ ' = Method.sign('' $\delta_i^*$ '', ' $SK_i^*$ '');
  statu_code = function.unSignCrypted( $\sigma_i, PK_i$ );
} else {
  statu_code = false;
}
return statu_code;

```

ALGORITHM 4: Authentication

the hash value δ_i^* . And the ciphertext C_i^{**} can be decrypted to acquire δ_i^{**} . After obtaining the above data, the next step will enter to the validation. There are two cases in this process.

Case 1. If $\delta_i^* = \delta_i^{**}$, the user can be authorized and gain the part access for userResource IR_i .

Case 2. If $\delta_i^* \neq \delta_i^{**}$, this user failed to authorization and output 0.

4.7. Conversation. Before two users establish a session, the system will set the maximum time limit for a single query to 300 s. After exceeding the time, it will be disconnected automatically and need to be verified again. During the verification process, account N_i^* will be queried whether exist. The following queries are same as those in Algorithm 5.

4.8. Transaction. The administrator can query the main parameters of the blockchain to get buildTime, buildType, genesisBlockHash and contractAddress, etc. These data are unique and cannot be tampered with arbitrarily.

```

Input:  $IR_i, SK_i$ 
Output: true or false
  Get cryptographic Method = new cryptographic
  (CryptoType.SCHNORRTYPE);
   $M_i = I_i$  ||  $PK_i$  ||  $IR_i$ ;
  ' $\delta_i$ ' || ' $\mu_i$ ' ||  $C_i$  ||  $\sigma_{IR_i} =$  function.SignCrypted( $M_i, SK_i$ );
  enter.set("' $IR_i, \sigma_{IR_i}$ '",  $IR_i, \sigma_{IR_i}$ );
  count = table.insert( $IR_i, \sigma_{IR_i}$ , entry);
  if (count == 1) {
    return true;
  } else {
    return false;
  }

```

ALGORITHM 5: Update.

5. Theoretical Analysis

5.1. Security Proof of Blockchain

5.1.1. Eclipse Attack. The multinode consortium blockchain system of TB-SCDM is built based on the FISCO BCOS platform. The system has a node access mechanism, so it is

```

Input:  $N_i$ ,  $PK_i$ 
Output:  $IR_i$ 
timeStamp = System.TimeSeconds();
expireTime = System.TimeSeconds() - timeStamp;
If(expireTime < 300) {
  Statu = select( $N_i$ );
  If(statu! = 0) {
    'PKi*' = result.getValue2();
    statu_code = function.unSignCrypted( $\sigma_{IR_i}$ , 'PKi*');
    If(statu_code == 1) {
      Return  $IR_i$ ;
    }
  }
}
}

```

ALGORITHM 6: Authorization.

```

Input:  $N_i$ ,  $PK_i$ 
Output: true or false
timeStamp = System.TimeSeconds();
expireTime = System.TimeSeconds() - timeStamp;
if(expireTime < 300) {
  statu = select( $N_i$ );
  if(statu != 0) {
    'PKi*' = result.getValue2();
    statu_code = function.unSignCrypted( $\sigma_i$ , 'PKi*');
    if(statu_code == 1) {
      creat.Conversation( $N_i$ );
      return true;
    }
  }
}
}

```

ALGORITHM 7: Conversation

difficult for attackers to obtain legitimate nodes through normal channels. Therefore, it is difficult for attackers to obtain legal nodes through normal channels. The PBFT mechanism of the TB-SCDM determines that if one third of the nodes of the system operate normally, it will not affect the normal operation of the whole system. Even if the attacker obtains the permissions of multiple accounting nodes, then the attacked node will be quickly discovered and processed by the central node.

5.1.2. DOS/DDoS Prevention. TB-SCDM adopts the consensus algorithm mechanism of consortium blockchain and PBFT. Therefore, the attack on ordinary nodes without accounting permission cannot hinder the normal operation of the blockchain system. Due to the characteristics of PBFT consistency algorithm mechanism, as long as there are more than one-third of normal nodes in the system, the system can operate normally, which leads to a huge inverse ratio between the attack cost and benefit of DDoS/DOS. However, for the consortium blockchain of TB-SCDM, the time and

cost of discovering and repairing accounting nodes are very small.

5.1.3. 51% Attack Prevention. For the consortium blockchain, the greater the computing power of all nodes, the more difficult to implement 51% attacks. It is hard for attackers to break more than 51% of nodes in a short time, and it is difficult to complete the destruction of the ledger before the central node takes corresponding countermeasures. Even if the ledger is attacked, the central node can repair the ledger in a very short time.

5.1.4. Sybil Attack Prevention. Each registered user will generate a unique public-private key pair. Each node needs a unique and unforgeable public key when uploading or updating the data on blockchain. Therefore, any attacker cannot use a single forged public key to disguise as multiple users and occupy all links of a billing node.

5.2. Security Proof of Signcryption

5.2.1. Identity Authentication. The system binds the user's public key with the user ID and then provides it to the user for safekeeping in the user registration stage. In addition, the signcryption method bound with the user's public key is adopted in the process of chaining or reading all information, which ensures the traceability of the system to the data and the authentication of the identity.

5.2.2. Confidentiality. Compared with the traditional digital signature, this paper adopts the signcryption technology based on Schnorr. Many literatures have verified the IND-CCA security (i.e., indistinguishability under the adaptive chosen-ciphertext attacks) based on Schnorr under the random oracles or standard oracles. Through the analysis of provable security theory, signcryption technology can effectively ensure the confidentiality of information in the process of transmission.

5.2.3. Unforgeability. TB-SCDM verifies whether the transmitted message comes from the real sender by verifying the message digest of the sender and receiver. We generate the compared message digest by storing the public key and trusted data in the blockchain. If the message digest is the same as the sender's message digest, verification can be realized to achieve UF-CMA security (i.e., existentially unforgeable under the adaptive chosen-message attacks). This article innovatively integrates signcryption, timestamp, and blockchain based on Schnorr to ensure the unforgeability of information.

6. Benchmark Test

6.1. Benchmark Test of Blockchain. In order to efficiently perform operations, we accessed the data on TB-SCDM using the CRUD interface supplied by FISCO BCOS 2.0. The hardware environment is an Intel i5-8265U 1.80 GHz computer, 16GB of memory, and running Windows 10 operating system.

It is available to deploy several different nodes on the same server for a test chain, we used a Linux server to deploy


```

Input: getchainVersion
Output: buildTime, buildType, genesisHash, etc.
[group:1]> getNodeVersion
ClientVersion{
  version='2.8.0',
  supportedVersion='2.8.0',
  chainId='1',
  buildTime='20210830 12:52:15',
  buildType='Linux/clang/Release',
  gitBranch='HEAD',
  genesisHash='bf0e0242a8040ead7549de49423712233a36d1b51b056a1c20df5eb78a9613e5'
}
transaction hash: 0xe88c2b9bf6dec9fa10356fd75b3d5414a5bd48f7ca246a8134e7f877928c47fc
contract address: 0x48102a5d29a6109384cb5a9c97d9fd07dd1a4416
currentAccount: 0xb13d80305a847dd2160c71465b50a6a1c0506ee3
[group:1]> getBlockNumber
9
[group:1]> getCurrentAccount
0xb13d80305a847dd2160c71465b50a6a1c0506ee3
    
```

ALGORITHM 8: Transaction

TABLE 2: The performance metrics of send rate, latency, and throughput.

Name	Succ	Fail	Send rate (TPS)	Max latency(s)	Min latency(s)	Avg latency(s)	Throughput (TPS)
User	1000	0	606.2	2.16	0.32	1.52	371.6
Transfer	10000	0	976.6	18.35	1.33	12.25	509.7

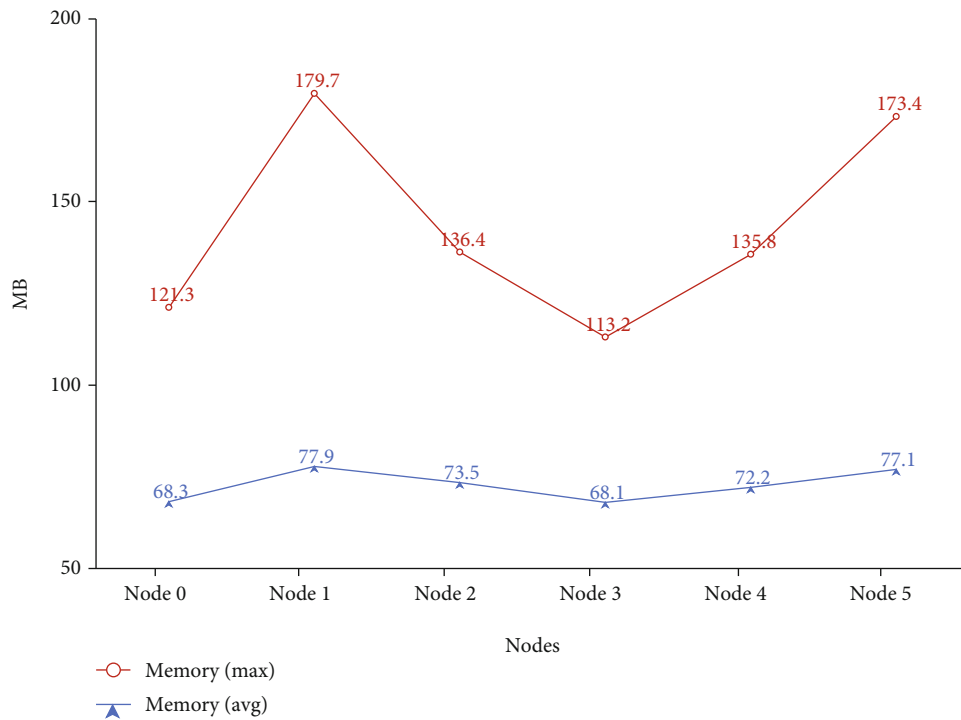


FIGURE 5: Memory usage of each node.

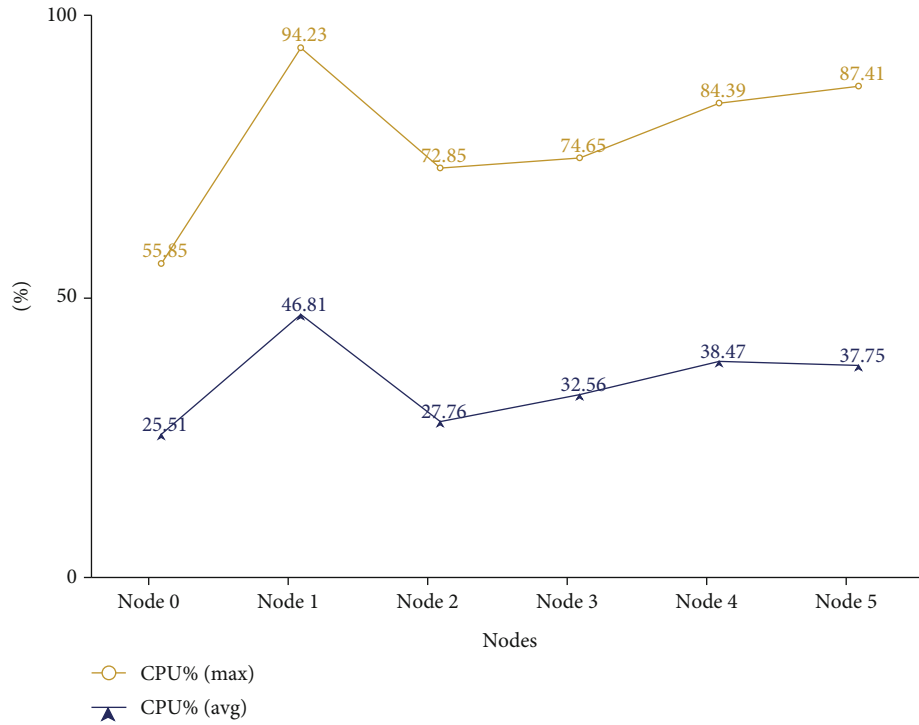


FIGURE 6: CPU usage of each node.

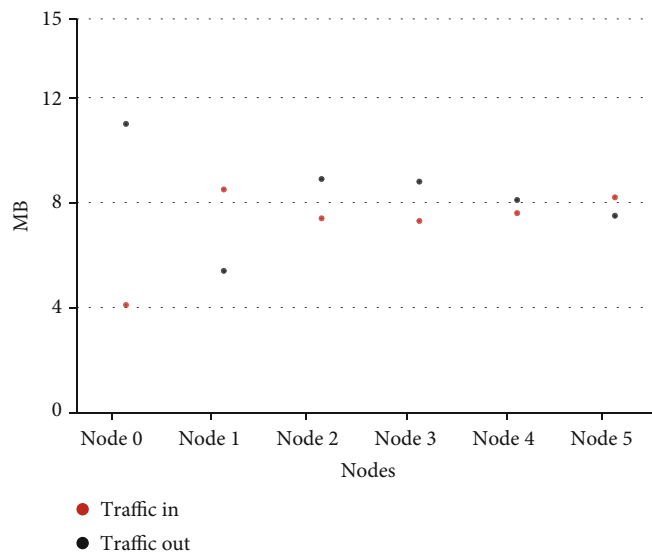


FIGURE 7: Traffic required for each node.

six nodes. For the smart contract of the blockchain, we chose the solidity language. This paper adopts Caliper as the test script to test the smart contract of consortium blockchain. The consortium blockchain is composed of a single group of six nodes. We select the scenario of 10000 concurrent transactions and 1000 new user registrations. The performance objects tested include memory usage, CPU usage, data traffic, disk read and write volume of each node, etc.

The performance metrics of send rate, latency, and throughput are described in Table 2. Figure 5 shows the memory usage of each node when processing data. Figure 6 shows the CPU usage of each node when verifying information. Figure 7 shows the traffic required for each node to form a consensus. Figure 8 shows the amount of traffic required by each node to form a consensus on the hard disk.

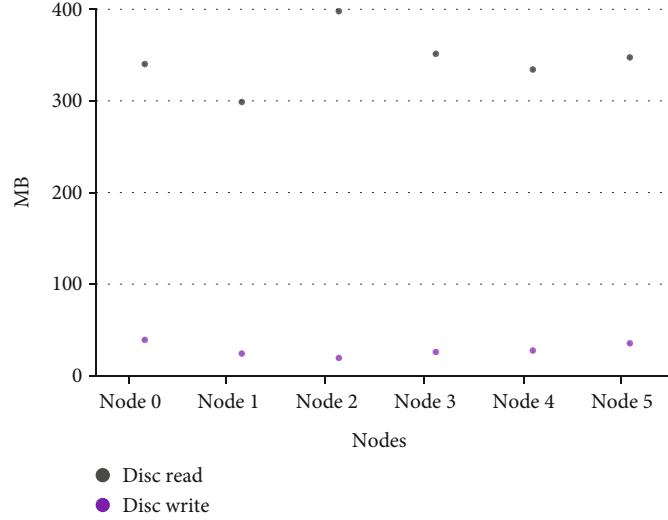


FIGURE 8: Disc read and write amount of each node.

TABLE 3: Symbols and descriptions of various operational times.

Symbols	Meaning
O_M	The time of a multiplication operation, $1 O_M \approx 1.25$ ms
O_P	The time of a bilinear pairing operations, $1 O_P \approx 32.23$ ms

TABLE 4: Performance comparison with other schemes.

Schemes	Signcryption cost	Unsigncryption cost	Execution time/($n = 1000$)	Confidentiality	Unforgeability
Iqbal et al. [28]	$4nO_M$	$nO_M + nO_P$	$5nO_M + nO_P/71980$ ms	✓	✓
Cui et al. [29]	nO_P	$2nO_M + 2nO_P$	$3nO_M + 2nO_P/166090$ ms	✓	✓
Hong et al. [30]	$n(3O_M + O_P)$	$n(3O_M + O_P)$	$6nO_M + 2nO_P/8820$ ms	✓	✓
Du et al. [31]	$4nO_M$	$4nO_M$	$8nO_M/8820$ ms	✓	✓
TB-SCDM	$2nO_M$	nO_M	$3nO_M/2520$ ms	✓	✓

6.2. *Benchmark Test of Signcryption.* The TB-SCDM and previous schemes [28–31] are exploited by the jPBC library on a laptop, where the configuration is a Windows 11 operating system, 2.60 GHz Intel(R) Core(TM) i7-9750H CPU with 16-GB RAM.

The meaning of the operation symbols is described in Table 3. The performance comparison of different schemes is described in Table 4.

A simple and intuitive method can be adopted in order to estimate the computation efficiency of the computational of several schemes. In terms of overall cryptographic operations, we can find that Iqbal et al. [28] is $5nO_M + nO_P$, Cui et al. [29] is $3nO_M + 2nO_P$, Hong et al. [30] is $6nO_M + 2nO_P$, Du et al. [31] is $8nO_M$, and TB-SCDM is $3nO_M$. From the perspective of formula, the cost efficiency of TB-SCDM is the highest.

Figures 9 and 10 describe the execution time of different schemes when n changes from 100 to 1000. From the perspective of change range, it can be seen that when the number of users gradually increases, the computational efficiency of TB-SCDM is more obvious than other schemes.

In Figure 11, in order to compare various schemes more clearly, we specially select the execution time of signcryption, unsigncryption, and total operations when the number of users n equals 1000. The execution times of signcryption operations are as follows: the running time of Iqbal et al. [28] is $4 \times 1000 \times 1.25 = 5000$ ms, the running time of Cui et al. [29] is $1000 \times 32.23 = 32230$ ms, the running time of Hong et al. [30] is $1000 \times 3 \times 1.25 + 1000 \times 32.23 = 35980$ ms, the running time of Du et al. [31] is $4 \times 1000 \times 1.25 = 5000$ ms, and the running time of TB-SCDM is $2 \times 1000 \times 1.25 = 2500$ ms.

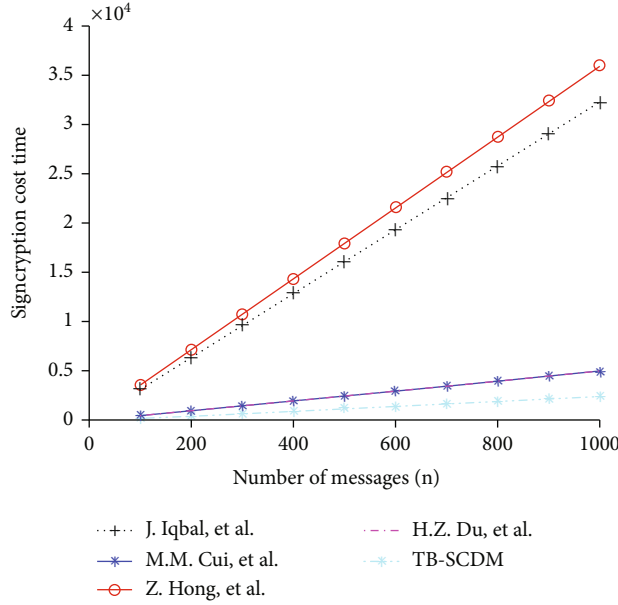


FIGURE 9: Comparison of signcrypton cost.

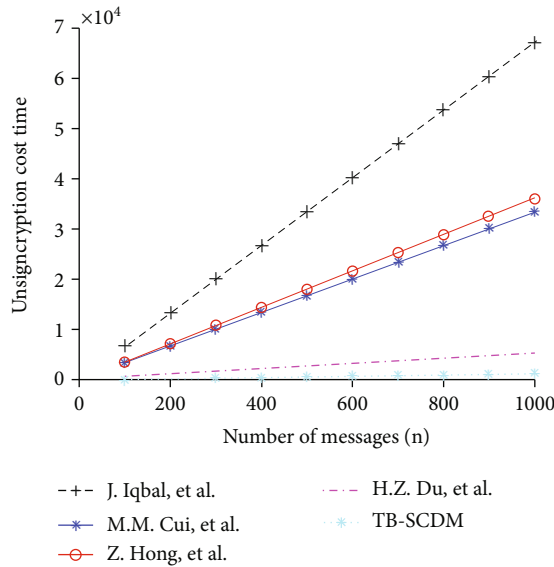


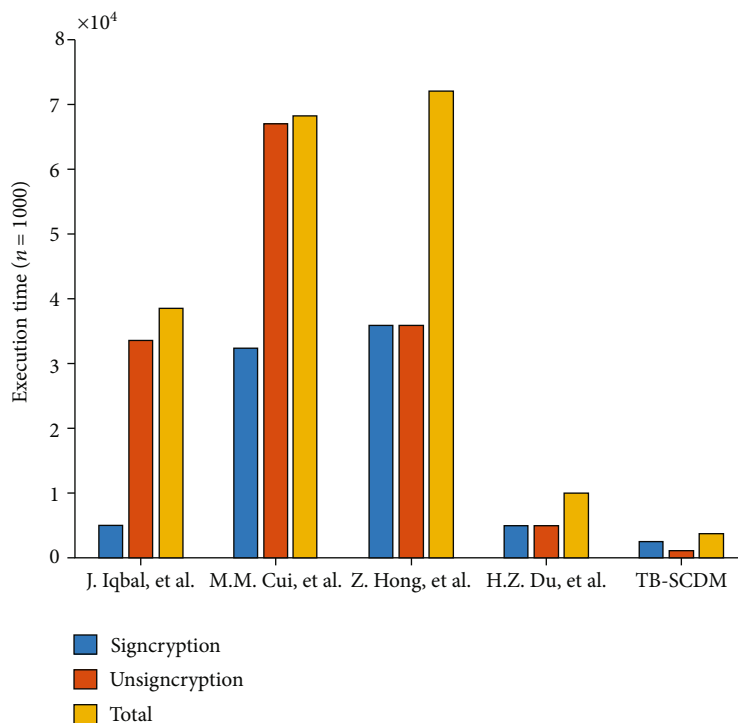
FIGURE 10: Comparison of unsigncrypton cost.

The execution times of unsigncrypton operations are as follows: the running time of Iqbal et al. [28] is $1000 \times 1.25 + 1000 \times 32.23 = 33480$ ms, the running time of Cui et al. [29] is $2 \times 1000 \times 1.25 + 2 \times 1000 \times 32.23 = 66960$ ms, the running time of Hong et al. [30] is $1000 \times 3 \times 1.25 + 1000 \times 32.23 = 35980$ ms, the running time of Du et al. [31] is $4 \times 1000 \times 1.25 = 5000$ ms, and the running time of TB-SCDM is $1000 \times 1.25 = 1250$ ms.

The execution time of total operations are as follows: the running time of Iqbal et al. [28] is $5 \times 1000 \times 1.25 + 1000$

$\times 32.23 = 38480$ ms, the running time of Cui et al. [29] is $3 \times 1000 \times 1.25 + 2 \times 1000 \times 32.23 = 68210$ ms, the running time of Hong et al. [30] is $6 \times 1000 \times 1.25 + 2 \times 1000 \times 32.23 = 71960$ ms, the running time of Du et al. [31] is $8 \times 1000 \times 1.25 = 10000$ ms, and the running time of TB-SCDM is $3 \times 1000 \times 1.25 = 3750$ ms.

On the whole, the computational efficiency of TB-SCDM is faster than the other four schemes [28–31]. In terms of security and algorithm efficiency, TB-SCDM is very suitable for secure communication in VANETs.

FIGURE 11: Comparison of total time ($n = 1000$).

7. Summary

In VANETs, SV using automatic driving need to access each other or RSU, GPS and other nodes to obtain reliable and stable data transmission services. Because VANET uses wireless communication, its openness allows attackers to easily obtain communication signals and further forge user nodes or Internet of Things nodes, which poses a greater security threat to SV. Based on the above reasons, this paper proposes a new trusted blockchain-based signaling protocol and data management for authentication and authorization. This scheme can effectively reduce the storage space occupied by information and the cost of signcryption verification.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Jinqi Su and Runtao Ren have contributed equally to this work and should be considered co-first authors.

Acknowledgments

This work is supported by the Xi'an Soft Science Research Project under the grant 2021-0019; National Training Program of Innovation and Entrepreneurship for Undergraduates under

the grant 202111664045; General Project of Humanities and Social Sciences Fund of the Ministry of Education (Youth Project) under the grant 20YJC630086; the Major Theoretical and Practical Research Project from Shaanxi Federation of Social Sciences Circles under the grant 20ZD195-144; the Research Grants Council of the Hong Kong Special Administrative Region, China (Project: CityU 11507219); the CityU SRG (Project: 7005780); Key Research Project on Major Theoretical and Practical Problems in Social Science Circles of Shaanxi Province under the grant SX-318; and Communication Soft Science Project of the Ministry of Industry and Information Technology under the grant R45.

References

- [1] J. Feng, Y. Wang, J. Wang, and F. Ren, "Blockchain-based data management and edge-assisted trusted cloaking area construction for location privacy protection in vehicular networks," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2087–2101, 2021.
- [2] F. Qu, Z. Wu, F. -Y. Wang, and W. Cho, "A security and privacy review of VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2985–2996, 2015.
- [3] R. Y. K. Lau, "Toward a social sensor based framework for intelligent transportation," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–6, Macau, China, 2017.
- [4] M. Arif, G. Wang, M. Zakirul Alam Bhuiyan, T. Wang, and J. Chen, "A survey on security attacks in VANETs: communication, applications and challenges," *Vehicular Communications*, vol. 19, article 100179, 2019.

- [5] A. Festag, "Standards for vehicular communication—from IEEE 802.11p to 5G," *e & i Elektrotechnik und Informationstechnik*, vol. 132, no. 7, pp. 409–416, 2015.
- [6] M. Sepulcre, M. Gonzalez-Martín, J. Gozalvez, R. Molina-Masegosa, and B. Coll-Perales, "Analytical models of the performance of IEEE 802.11p vehicle to vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 713–724, 2022.
- [7] A. T. Giang, A. Busson, A. Lambert, and D. Gruyer, "Spatial capacity of IEEE 802.11p-based VANET: models, simulations, and experimentations," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6454–6467, 2016.
- [8] M. Dibaei, X. Zheng, Y. Xia et al., "Investigating the prospect of leveraging blockchain and machine learning to secure vehicular networks: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 683–700, 2022.
- [9] C. Y. Yeung, L. C. K. Hui, T. W. Chim, S. Yiu, G. Zeng, and J. Chen, "Anonymous Counting Problem in Trust Level Warning System for VANET," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 34–48, 2019.
- [10] S. Latif, S. Mahfooz, N. Ahmad et al., "Industrial Internet of Things based efficient and reliable data dissemination solution for vehicular ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 2018, no. 1, pp. 1–16, 2018.
- [11] A. Khalid, M. S. Ifikhar, A. Almogren, R. Khalid, M. K. Afzal, and N. Javaid, "A blockchain based incentive provisioning scheme for traffic event validation and information storage in VANETs," *Information Processing & Management*, vol. 58, no. 2, article 102464, 2021.
- [12] W. Luo and W. Ma, "Efficient and secure access control scheme in the standard model for vehicular cloud computing," *IEEE Access*, vol. 6, pp. 40420–40428, 2018.
- [13] Z. Lu, Q. Wang, G. Qu, H. Zhang, and Z. Liu, "A blockchain-based privacy-preserving authentication scheme for VANETs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 12, pp. 2792–2801, 2019.
- [14] Y. Yang, L. Zhang, Y. Zhao, K. K. R. Choo, and Y. Zhang, "Privacy-preserving aggregation-authentication scheme for safety warning system in fog-cloud based VANET," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 317–331, 2022.
- [15] R. Guo, G. Yang, H. Shi, Y. Zhang, and D. Zheng, "O3-R-CP-ABE: an efficient and revocable attribute-based encryption scheme in the cloud-assisted IoMT system," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8949–8963, 2021.
- [16] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.
- [17] Q. Wang, R. Y. K. Lau, and X. Mao, "Blockchain-enabled smart contracts for enhancing distributor-to-consumer transactions," *IEEE Consumer Electronics Magazine*, vol. 8, no. 6, pp. 22–28, 2019.
- [18] Y. Zhong and W. Wu, "A switching-based interference control for booster separation of hypersonic vehicle," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5560621, 9 pages, 2021.
- [19] X. Liu, H. Huang, F. Xiao, and Z. Ma, "A blockchain-based trust management with conditional privacy-preserving announcement scheme for VANETs," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4101–4112, 2020.
- [20] B. Luo, X. Li, J. Weng, J. Guo, and J. Ma, "Blockchain enabled trust-based location privacy protection scheme in VANET," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2034–2048, 2020.
- [21] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, and W. He, "An efficient decentralized key management mechanism for VANET with blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5836–5849, 2020.
- [22] I. Dohare, K. Singh, A. Ahmadian, S. Mohan, and P. K. Reddy M, "Certificateless aggregated signcryption scheme for Cloud-Fog Centric Industry 4.0," *IEEE Transactions on Industrial Informatics*, p. 1, 2022.
- [23] L. Jiang, T. Li, X. Li, M. Atiquzzaman, H. Ahmad, and X. Wang, "Anonymous communication via anonymous identity-based encryption and its application in IoT," *Wireless Communications and Mobile Computing*, vol. 2018, 8 pages, 2018.
- [24] W. Luo and W. Ma, "Secure and efficient data sharing scheme based on certificateless hybrid signcryption for cloud storage," *Electronics*, vol. 8, no. 5, pp. 590–601, 2019.
- [25] X. Ye, G. Xu, X. Cheng, Y. Li, and Z. Qin, "Certificateless-based anonymous authentication and aggregate signature scheme for vehicular ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 2021, 16 pages, 2021.
- [26] M. Lefebvre, S. Nair, D. W. Engels, and D. Horne, "Building a Software Defined Perimeter (SDP) for network introspection," in *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 91–95, Heraklion, Greece, 2021.
- [27] K. Lei, Q. Zhang, L. Xu, and Z. Qi, "Reputation-based byzantine fault-tolerance for consortium blockchain," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 604–611, Singapore, 2018.
- [28] J. Iqbal, A. I. Umar, N. Amin, and A. Waheed, "Efficient and secure attribute-based heterogeneous online/offline signcryption for body sensor networks based on blockchain," *International Journal of Distributed Sensor Networks*, vol. 15, no. 9, Article ID 155014771987565, 2019.
- [29] M. Cui, D. Han, J. Wang, K. -C. Li, and C. -C. Chang, "ARFV: an efficient shared data auditing scheme supporting revocation for fog-assisted vehicular ad-hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15815–15827, 2020.
- [30] Z. Hong, F. Tang, and W. Luo, "Privacy-preserving aggregate signcryption for vehicular ad hoc networks," in *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy*, pp. 72–76, New York: ACM Press, 2018.
- [31] D. U. Hongzhen, W. E. Qiaoyan, Z. H. Shanshan, and G. A. Mingchu, "A pairing-free certificateless signcryption scheme for vehicular ad hoc networks," *Chinese Journal of Electronics*, vol. 30, no. 5, pp. 947–955, 2021.