Efficient and Automatic Pseudonym Management Scheme for VANET with Blockchain

Xiangsong Zhang[1], Ming Yuan[2], and Zhenhua Liu[2,3](✉)

[1] Xi'an Technological University, Xi'an 710021, China
[2] School of Mathematics and Statistics, Xidian University, Xi'an 710071, China
zh_liu@mail.xidian.edu.cn
[3] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China

Abstract. As a product of the development of intelligent transportation system, vehicular ad hoc network (VANET) has been widely studied in recent years, where the vehicles would utilize many pseudonyms to achieve conditional privacy protection. With the increasing number of pseudonyms, the management of pseudonyms would become a new challenge. In this paper, we investigate pseudonym management in VANET based on blockchain. Firstly, an efficient and automatic pseudonym management scheme is proposed to realize the registration, update and revocation of vehicle's pseudonyms. At the same time, the voting system is applied to the pseudonym revocation protocol, which can provide a solution of the legal vehicle's pseudonym being revoked wrongly. Then, security analysis shows that the proposed scheme can meet the security requirements of VANET. Finally, the performance of the proposed scheme is analyzed through the experiments and simulations. The experimental results show that the automatic pseudonym management scheme is practical and superior to the existing schemes in terms of storage and computational overhead.

Keywords: VANET · Blockchain · Pseudonym management · Smart contract

1 Introduction

Recent advances in wireless communication technologies and automobiles have fueled the growth of intelligent transport system (ITS) that can address various vehicular traffic issues, such as traffic congestions, information disseminations, and accidents. VANET is an integral component of ITS, where the moving vehicles are connected and communicated by wireless [15]. Each participating vehicle is equipped with an on-board unit (OBU) that can communicate with nearby

Supported by by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant No. 2022JZ-38 and the National Natural Science Foundation of China under Grants No. 61807026.

E. Ahene and F. Li (Eds.): FCS 2022, CCIS 1726, pp. 211–225, 2022.
https://doi.org/10.1007/978-981-19-8445-7_14

vehicles and roadside units (RSUs). Furthermore, RSUs can connect to the backbone network for data exchange or sharing via the Internet [12].

In such an open access VANET, the vehicle communication system is vulnerable, which would threaten the privacy of vehicles [17]. The technical specifications represented by IEEE WAVE 1609.2 [5] and ZETSI 102 [6] have proposed some security and privacy solutions. For instance, vehicle public key infrastructure (VPKI) can provide multiple short-term certificates (pseudonymes) for legitimate vehicles [16], which can switch from one pseudonym to another to realize the unlinkability. If a vehicle's pseudonym certificate expires, the pseudonym and the certificate need to be revoked. Therefore, pseudonym management becomes one of serious security issues in VANET [18]. Currently, there are many solutions to pseudonym management based on PKI technology, which can realize authentication and anonymity [2,11,13,19,20,23,26–28].

Furthermore, these schemes from PKI adopted the centralized management model, i.e. Trust Authority (TA). Obviously, there are some limitations, such as the single point of failure [1], the massive communication overhead [2], and the false revocation [17]. Recently, based on a distributed platform blockchain, some key management schemes [8,9,14] had been proposed for VANET. Unfortunately, these schemes cannot support the automatic key management. Specially, Lei et al.'s scheme [9] cannot support key update, and the other schemes [8,14] cannot realize the key agreement that is a key protocol to protect data transmission directly. Actually, the blockchain technology can be viewed as a double-edged sword for key management in VANET. On one hand, the tamper-proof property of blocks can be used to construct a trust chain for public key. On the other hand, the property brings some troubles on key management such as key update and revocation. In order to realize efficient key management in VANET, Ma et al. [15] used smart contract [21] to manage the vehicles' key in an automatic way and proposed a decentralized key management mechanism based on blockchain. Thus blockchain technology would be applied to key management that gives a new direction to pseudonym management in VANET.

In order to realize efficient and secure pseudonym management for VANET, we propose an automatic pseudonym management scheme. The major contributions of the paper are as follows.

- **Distributed storage of pseudonym**. The decentralized pseudonym management scheme frees VANET from the dependence on PKI by using the blockchain-based tamper-proof and distributed storage of pseudonym. The distributed storage based on blockchain is suitable for a distributed VANET, and also makes VANET more robust against the single point of failure compared with the existing centralized models. In addition, since each RSU can know the anonymity of each vehicle through the blockchain, cross-regional anonymous authentication can be achieved among the vehicles.
- **Automatic pseudonym management**. Based on the smart contract technique, the decentralized pseudonym management scheme can implement automatic registration, update and revocation of the vehicle's pseudonyms. At

the same time, the application of smart contract can improve the efficiency of pseudonym management.
- **Pseudonym update and revocation**. The blockchain technique is applied to accelerate the dissemination of the updated or revoked pseudonym to the entire network. The decentralized pseudonym management scheme can employ a blockchain and smart contract based on the decentralized voting mechanism to detect some malicious vehicles with adversarial behaviors.

2 System Framework

2.1 System Model

As shown in Fig. 1, an automatic pseudonym management scheme includes three entities: Register authority (RA) , blockchain network [15] and on-board unit.

- **RA**. The tasks include the deployment of blockchain network and smart contract, the issuance of transaction data, and the verification of vehicle identity. RA is also responsible for generating the initial pseudonym for vehicle.
- **OBU**. As a processing unit embedded in the vehicle, OBU is responsible for V2V and V2I communications. In addition, a hardware security module is installed to securely store cryptographic materials. Through OBU during driving, all vehicles can regularly send some security information that consists of the driving speed, the driving direction, the vehicle position, and so on. Furthermore, the security information is collected by RSUs.
- **Blockchain network**. It is a peer-to-peer network constructed by RSUs. Each RSU sends the transactions and runs the mining function. Blockchain network can accelerate the transactions and the synchronization of blocks. As the miners, some RSU nodes need to use proof of work and proof of stake consensus mechanism to create new blocks. In addition, RSUs can generate the pseudonyms, pseudonym certificates, as well as the corresponding public and private keys for the legal vehicles according to the anonymous credentials submitted by the vehicles. When a pseudonym is revoked, RSU is responsible for distributing the revocation information of pseudonym certificate.

2.2 Attack Model

Assume that RA, RSUs and vehicles are equipped with hardware security modules that are responsible for securely storing cryptographic materials, and RA is honest-but-curious and executes faithfully any programming protocol.

In a pseudonym management scheme there exist two types of attacks, i.e. internal attacks and external attacks [15]. Internal attacks can be performed by an adversary Adv_1, whose goals are to decrypt the ciphertexts to obtain the other vehicles' private data and services with the following capabilities:

1. Adv_1 can eavesdrop on all communications in VANET to obtain the encrypted data.

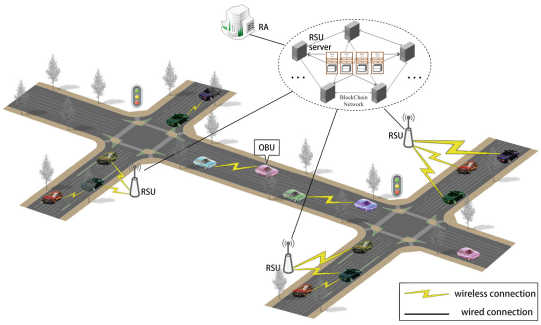

Fig. 1. System model

2. Adv_1 can compromise RSUs to guess the traffic contents between the vehicles and RSUs.

On the other hands, an adversary Adv_2 can execute the external attacks, and enable the unauthorized users to login, enjoy services and even destroy VANET with the following capabilities:

1. Adv_2 can eavesdrop on all communication in VANET to obtain the encrypted data and guess the plaintexts.
2. Adv_2 can compromise the vehicle or RSU to guess the legal identity.
3. Adv_2 can tamper or masquerade the messages from the legal participants.

2.3 Design Goals

We will propose an efficient and automatic pseudonym management scheme for VANET by using of blockchain. According to the security requirements and attack model, the proposed scheme should satisfy the following design goals:

- **Distributed storage and automatic management**. Our scheme can utilize blockchain to support the distributed storage of the vehicle's pseudonyms, and use smart contract to automatically manage these pseudonyms [15].
- **Authentication**. The proposed scheme can support the authentication between RSU and OBU by negotiating a shared session key. After successful authentication, a secure channel for communication will be created [15].
- **Pseudonym update and revocation**. When a pseudonym is expired, the proposed scheme can use smart contract to update the vehicle's pseudonym. Furthermore, our scheme can realize the pseudonym revocation by using the decentralized voting mechanism on smart contract [17].

3 Efficient and Automatic Pseudonym Management Scheme

3.1 System Setup Phase

RA executes the initialization and generates the public/secret keys PK_{RA}, SK_{RA}, and each RSU computes its public/secret keys PK_{RSU}, SK_{RSU}. A smart contract managing pseudonyms will be established on blockchain. RA creates a blockchain account for each blockchain network node through the account generation tools and then uses the account address to create a smart contract. After RA deploys the smart contract successfully, blockchain network will create automatically a contract address. Only RA and blockchain network nodes can send a transaction to trigger the execution of smart contract that offers four functions, including *RegisterPseudonym*, *UpdatePseudonym*, *VotePseudonym* and *RevokePseudonym*, to manage the pseudonyms of the vehicles.

3.2 Registration Phase

OBU holds a long-term certificate (LTC) containing identity information ID and completes the initial registration process with RA through a secure channel to access VANET. At the same time, OBU will receive IP_{cert_v} issued by RA. As shown in Table 1, OBU and RA execute the registration protocol as below.

- **Step 1:** OBU registers through a secure channel, and sends the long-term certificate and $[t_s, t_e]$ to RA, where $[t_s, t_e]$ is a pseudonym request interval.
- **Step 2:** RA encrypts OBU's identity ID to generate an initial pseudonym IP_v and the corresponding ipk_v, isk_v according to the system parameters. After that, a "credential identifiable key" ($IK_{IP_{cert_v}}$) is created to bind the credential to the vehicle's certificate: $IK_{IP_{cert_v}} = h(C||t_s||t_e||Rnd_{IK_{IP_{cert_v}}})$, where $C = Enc_K(IP_v, exp)$, $Rnd_{IK_{IP_{cert_v}}}$ is a random number generated by RA for this credential, and exp is the expiration of the long-term certificate. Then RA generates IP_{cert_v} that includes χ and $Sign_{SK_{RA}}(\chi)$, where $\chi \leftarrow (C, IK_{IP_{cert_v}}, t_s, t_e)$. RA sends IP_v, ipk_v, isk_v, IP_{cert_v}, $IK_{IP_{cert_v}}$, $Rnd_{IK_{IPcert_v}}$ to OBU through the secure channel.
- **Step 3:** OBU stores $IP_v, ipk_v, isk_v, IP_{cert_v}, IK_{IP_{cert_v}}, Rnd_{IK_{IPcert_v}}$.

3.3 Authentication Phase

After receiving IP_v, ipk_v, isk_v and $IPcert_v$, OBU interacts with RSU to execute the V2I authentication as shown in Table 2. Then a secure channel is established.

- **Step 1:** When OBU moves to the wireless communication range of the accessible RSU, the V2I authentication protocol will be executed. RSU randomly selects x_R, calculates $h_R = g^{x_R}$, and generates a signature $\sigma_1 = Sign_{SK_{RSU}}(h_R, TS_1)$, where TS_1 is the time-stamp. RSU periodically broadcasts $\{R_{ID}, \sigma_1, h_R, TS_1\}$.

Table 1. Registration protocol

OBU		RA
	Send $LTC, [t_s, t_e]$ →	
		Generate IP_v, ipk_v, isk_v
		Compute C, IK_{IPcert_v}
		Generate $IPcert_v$
	Send $IP_v, isk_v, IK_{IPcert_v}$, $IPcert_v$, $Rnd_{IK_{IPcert_v}}$ ←	
Store $IP_v, isk_v, IK_{IPcert_v}$, $ipk_v, IPcert_v, Rnd_{IK_{IPcert_v}}$		

- **Step 2:** After receiving the broadcasting messages, OBU first checks whether TS_1 is fresh. If TS_1 is fresh, OBU uses R_{ID} to verify σ_1. If the verification is successful, OBU selects randomly y_V, computes $h_V = g^{y_V}$, and generates a signature $\sigma_2 = Sign_{isk_v}(h_V, TS_2)$, where TS_2 is the time-stamp.. Then OBU calculates a shared key $K_{V-R} = {h_R}^{y_V}$ with RSU. OBU uses K_{V-R} to generate $c = Enc_{K_{V-R}}(IP_v)$. OBU sends c, σ_2, h_V, TS_2 to RSU.
- **Step 3:** After receiving the data, RSU checks if TS_2 is fresh. If it is fresh, RSU calculates the shared key $K_{V-R} = {h_V}^{x_R}$, uses K_{V-R} to decrypt c to obtain IP_v, and uses IP_v to verify σ_2. If the verification holds, OBU is regarded as a legal one. Otherwise, RSU will reject the access request from OBU.

If the verification is successful, RSU and OBU can establish a secure channel by negotiating a shared key that is created by Diffie-Hellman key agreement.

Table 2. V2I authentication protocol

RSU		OBU
Select x_R, compute h_R		
Generate σ_1	Send $R_{ID}, \sigma_1, h_R, TS_1$ →	
		Check TS_1, verify σ_1
		Compute h_V, generate σ_2
		Compute K_{V-R}, generate c
	Send c, σ_2, h_V, TS_2 ←	
Check TS_2, compute K_{V-R},		
Decrypt c, obtain IP_v		
Verify σ_2		

3.4 Pseudonyms Generation Phase

By using the secure channel, RSU sends the pseudonyms, the pseudonym certificates, and the corresponding public and secret keys to OBU. As shown in Table 3, the steps of pseudonym generation protocol are listed as follows.

- **Step 1:** OBU generates a pseudonym request message $m = (Rnd_{IK_{IPcert_v}}, IPcert_v, [t'_s, t'_e])$, where t'_s and t'_e are the start time-stamp and the end time-stamp of the actual pseudonym request interval. Then OBU sends $\{ID_{req}, m, n, TS_3\}$ to RSU, where ID_{req} is the pseudonym request identity, n is a freshly random value, and TS_3 is the time-stamp.
- **Step 2:** After receiving the request, RSU first uses the shared key with OBU to decrypt the request message and verifies the validity of IP_{cert_v}: $Verify_{PK_{RA}}(IP_{cert_v})$. If OBU's credential is valid, RSU checks whether the actual period of the requested pseudonyms (i.e., $[t'_s, t'_e]$) is within the period specified in the credential (i.e., $[t_s, t_e]$) and OBU indeed has the credential by verifying if the equation $IK_{IP_{cert_v}} = h(C||t_s||t_e||Rnd_{IK_{IPcert_v}})$ holds. RSU chooses random numbers to generate public/secret keys (pk_v^i, sk_v^i) and the corresponding public key certificates $Cert_v^i$ $(i = 1, \cdots, n)$ for OBU, where n is the number of pseudonyms distributed by RSU each time, and the public key certificates are signatures generated by RSU with its secret key SK_{RSU}. Then RSU generates "pseudonym identifiable key" $IK_{P_v^i}$ to bind pseudonyms to OBU's credential: $IK_{P_v^i} = h(IK_{IP_{cert_v}}||pk_v^i||t_s^i||t_e^i||h^i(Rnd_v))$. RSU implicitly associates a batch of pseudonyms belonging to each OBU by calculating the pseudonym sequence number SN, i.e., $SN^1 = h(IK_{P_v^1}||h^1(Rnd_v))$, and $SN^i = h(SN^{i-1}||h^i(Rnd_v)), i = 2, \cdots, n$. Afterwards RSU generates pseudonyms for OBU: $P_v^i \leftarrow (SN^i, IK_{P_v^i}, t_s^i, t_e^i)$, $PS = \{(P_v^1, pk_v^1, sk_v^1, Cert_v^1), \cdots, (P_v^n, pk_v^n, sk_v^n, Cert_v^n)\}$. RSU encapsulates the binding data $\{IP_{cert_v}, PS\}$ in JSON format and then encodes as hexadecimal embedded into the data field of the transaction. Then RSU sends the transaction to blockchain network and triggers the smart contract function *RegisterPseudonym*. After smart contract is executed and the mining is successful, the transaction record is added to the blockchain. RSU sends $\{ID_{res}, PS, Rnd_v, n+1, TS_4\}$ to OBU by a secure channel, where ID_{res} is a pseudonym response message.
- **Step 3:** After receiving the response message from RSU, OBU first recovers the message with the shared key, and then verifies $IK_{P_v^i}$ by verifying whether the equation $h(IK_{IP_{cert_v}}||pk_v^i||t_s^i||t_e^i||h^i(Rnd_v)) == IK_{P_v^i}$ holds. If the verification is successful, OBU stores PS.

Table 3. Pseudonym generation protocol

OBU		RSU
Prepare m	Send ID_{req}, m, n, TS_3 $\longrightarrow$	
		Verify IP_{cert_v}, check $IK_{IP_{cert_v}}$ Generate Rnd_v, compute P_v^i Generate PS, trigger SC
	$\longleftarrow$ Send $ID_{res}, PS, Rnd_v, n+1, TS_4$	
Verify $IK_{P_v^i}$ Store PS		

3.5 Pseudonyms Update Phase

The pseudonyms update requires RSU to issue new pseudonyms by using of smart contracts. If the pseudonym request time in the credential is about to expire, OBU sends the current IP_{cert_v} and the new request time interval $[t'_s, t'_e]$ to RA to apply for a new credential through RSU. After RA validates IP_{cert_v}, a new credential is generated to replace IP_{cert_v} that will soon be unavailable. As shown in Table 4, the steps are described in detail as follows.

- **Step 1:** OBU sends $\{\sigma_3, IP_{cert_v}, h_V, TS_5\}$ to the nearby RSU, where $\sigma_3 = Sign_{isk_v}(IP_{cert_v}, h_V, TS_5)$.
- **Step 2:** After receiving $\{\sigma_3, IP_{cert_v}, h_V, TS_5\}$, RSU temporarily saves h_V and then forwards them from OBU to RA.
- **Step 3:** After getting the data, RA checks the validity of IP_{cert_v} and verifies σ_3. If the verifications are both successful, RA generates a new credential $IP_{cert'_v}$ for OBU by selecting a new $Rnd_{IK_{IP_{cert'_v}}}$ and using its private key SK_{RA} to generate a signature, and returns $IP_{cert'_v}$, $Rnd_{IK_{IP_{cert'_v}}}$ to RSU.
- **Step 4:** After receiving the return message, RSU temporarily saves $IP_{cert'_v}$, $Rnd_{IK_{IP_{cert'_v}}}$, and sends $\{R_{ID}, \sigma_4, h_R, TS_6\}$ to OBU, where $h_R = g^{x_R}$, $\sigma_4 = Sign_{SK_{RSU}}(h_R, TS_6)$. Then, RSU calculates a shared key $K_{V-R} = {h_V}^{x_R}$.
- **Step 5:** After receiving $\{R_{ID}, \sigma_4, h_R, TS_6\}$, OBU first checks whether TS_6 is fresh. If TS_6 is fresh, OBU continues to verify σ_4. If the verification is successful, OBU calculates a shared key $K_{V-R} = {h_R}^{y_V}$.

After the above steps, OBU and RSU can establish a secure channel. RSU encapsulates the binding data $\{IP_{cert'_v}, PS'\}$ in JSON format and then encodes as hexadecimal embedded into the data field of the transaction. Then RSU sends the transaction to blockchain network and triggers SC function $UpdatePseudonym$. After smart contact is executed and the mining is successful, the transaction record is added to blockchain. RSU sends $\{IP_{cert'_v}, PS', Rnd_{IK_{IP_{cert'_v}}}\}$ to OBU through the secure channel.

3.6 Pseudonyms Revocation Phase

When OBU has been found to have some malicious behaviors in VANET, such as reading disloyal traffic information, the pseudonyms of OBU should be revoked in time. The process of pseudonym revocation is described in detail as follows.

- **Step 1:** If OBU_j receives a false message m from OBU_i, OBU_j will generates a *report*, including m, the pseudonym, and pseudonym certificate.
- **Step 2:** OBU_j sends the *report* to the nearest RSU. *RSU* checks whether the message m is malicious. If so, RSU encapsulates a voting transaction and sends to blockchain network for triggering smart contact and adding a ticket to $VotePseudonym_{OBU_i}$. If OBU_i continues to perform some malicious operations, RSU adds another ticket to $VotePseudonym_{OBU_i}$. Once the vehicle's $VotePseudonym_{OBU_i}$ exceeds a threshold Thr, i.e., $VotePseudonym_{OBU_i} \geq Thr$, smart contact will notify RSU.

– **Step 3:** RSU sends a revocation transaction to blockchain and triggers *RevokePseudonym* to remove OBU's pseudonym and certificate. In addition, RSU periodically checks the validity period of the unrevoked OBU's pseudonym and sends the transaction for triggering smart contact to remove the user pseudonym when it is expired. RSU broadcasts the revocation information, and further transfers it to RA. RA can directly recover the real identity of OBU through the decryption, and then revoke LTC of the vehicle.

In the proposed scheme, each RSU releases the revocation information at any time to notify the vehicles in any new revocation event. The vehicles can receive the latest certificate revocation list timely through RSUs.

Table 4. Pseudonym update protocol

OBU		RSU		RA
	Send $\sigma_3, h_V,$ IP_{cert_v}, TS_5 $\longrightarrow$			
		Save h_V		
			Send $\sigma_3, h_V,$ IP_{cert_v}, TS_5 $\longrightarrow$	
				Verify $\sigma_3, IP_{cert_v},$ h_V, TS_5
				Select $Rnd_{IK_{IPcert'_v}}$
				Generate $IP_{cert'_v}$
			Send $IP_{cert'_v},$ $Rnd_{IK_{IPcert'_v}}$ $\longleftarrow$	
		Save $Rnd_{IK_{IPcert'_v}},$ $IP_{cert'_v}$		
		Generate σ_4, h_R		
		Compute K_{V-R}		
	Send $R_{ID}, \sigma_4,$ h_R, TS_6 $\longleftarrow$			
Check TS_6				
Verify σ_4				
Compute K_{V-R}				
		Trigger SC		
	Send $IP_{cert'_v}, PS',$ $Rnd_{IK_{IPcert'_v}}$ $\longleftarrow$			
Store $IP_{cert'_v}, PS'$ $Rnd_{IK_{IPcert'_v}}$				

4 Security Analysis

We only discuss the ability of automatic pseudonym management scheme against typical attacks towards VANET.

1. **Resisting Internal Attacks:** Adv_1 eavesdrops on the communication data between OBU and RSU. However, since the data are encrypted by them, Adv_1 cannot decrypt the data without the session key. In addition, the session key is calculated securely between OBU and RSU. Furthermore, if Adv_1 wants to decrypt, it needs to obtain RSU's secret key, but these parameters are securely stored in HSM, and thus it is very difficult to realize the goal. Even if RSU is compromised, Adv_1 cannot obtain the secrets in HSM and affect the vehicle.
2. **Resisting External Attacks:** According to the attack model, we enumerate several important attacks, such as replay attacks, DoS attacks and collusion attacks, launched by Adv_2. These attacks can be prevented effectively.
   (a) **Replay Attack:** Adv_2 uses network listening or other means to steal the authentication credentials, and then re-send them to RSU. In the proposed scheme, OBU and RSU use in conjunction with nonce and time-stamp TS checking, which can effectively thwart replay attacks.
   (b) **DoS Attack:** Adv_2 is compromised by malicious organizations to act irrationally (e.g., initiate DoS attacks). Adv_2 sends a large number of intercepted message to VANET, which causes some legitimate requests to fail to respond. In the proposed scheme, we add time-stamps TS to ensure the freshness of the messages and prevent the expiring messages from Adv_2. In addition, it is impossible that Adv_2 compromises the legitimate users to launch DoS attacks, since the cryptographic materials of all legitimate users are protected by HSM.
   (c) **Collusion Attack:** Adv_2 can collude with the other compromised users to disrupt VANET or obtain the private data by stealing the session key. After the mutual authentication between OBU and RSU, RSU issues multiple anonymous identities and the corresponding signing keys to OBU. The keys and pseudonyms will be encrypted with the shared session key K_{V-R}, which effectively prevents the keys from being stolen by attackers during the key transmission. Key leakage can not occur at this time, thus the entire network is secure. Thus, the proposed scheme can defense collusion attacks in the actual VANET.

5 Performance Evaluation

5.1 Implementation and Gas Cost

To analyze the practicality of the proposed automatic pseudonym management scheme, a prototype of smart contact is compiled and deployed on the testnet of the Ethereum network, Rinkeby. Here, Rinkeby not only provides a free request of funds, but also designs a user friendly web interface for a convenient block explorer. Smart contact is deployed on the Rinkeby Testnet with the addresses:

– **RA's account address**:

$$0x8c29789a5017e77b9e00634536b288a9085a4d44,$$

– **RSU's account address**:

$$0xeec732d6b74f9354b8a12da9ace819418066918b.$$

The details of this implementation are presented as follows.

1. Firstly, we use MyEtherWallet to generate two accounts for our test, switch to RA's account, and request 3 Ethers from Rinkeby such that RA can publish the transactions.
2. Then, we execute the followings as RA's identity. We deploy the smart contract into Rinkeby using Remix. The creation of smart contact is only performed once and the cost is \$0.2576.
3. Next, we simulate RSU to add the anonymity of the vehicle to blockchain. We switch to RSU's account and trigger the smart contact functions *Register –Pseudonym* and *UpdatePseudonym*. The cost of *RegisterPseudonym* and *UpdatePseudonym* operation are \$0.0259 and \$0.0093, respectively.
4. Finally, RSU sends a revocation transaction to the blockchain and triggers *RevokePseudonym*. The cost of *RevokePseudonym* operation is \$0.0158.

The costs measured by the experiment are shown in Table 5. We compare the proposed scheme with PKI-based solutions [22,25]. According to [22], the initial setup cost of traditional PKI infrastructure is about \$10,000, and the annual management fee is about \$45,000. Assume that the proposed initial setup and annual overhead are the same as the traditional PKI. The approximate cost of managing a car based on a traditional PKI is \$20 per year, but the approximate cost of managing a car based on a blockchain is \$0.30 per year. The results in Fig. 2 show that Display POWER MANAGEMENT Signalling is practical in application.

Table 5. Costs of the different functions in the SC

Functionalities	Gas used	Actual cost (Ether)	USD
Smart Contract Creation	1302525	0.001302525	0.2579
RegisterPseudonym	130808	0.000130808	0.0259
UpdatePseudonym	46970	0.000046970	0.0093
RevokePseudonym	797979	0.000797979	0.0158

5.2 Storage Overhead

The storage overhead of the proposed scheme depends on the amount of vehicle and RSU storage pseudonyms. According to the IEEE standard [4], the size of the certificate is 126 bytes, the public key size of RSU is 29 bytes, and the pseudonym of vehicle is 32 bytes. By implementing the smart contract on Ethereum and

analyzing results, it is concluded that the size of a transaction data is 100 bytes, one block contains about 15 transactions, and the block header is about 200 bytes. Assuming there are 1 million vehicles in the network, the amount of data that needs to be stored by each RSU is 30 Mbytes. If there are 100000 RSUs deployed in the network, the amount of data that is maintained by each vehicle is equal to 2.8 Mbytes. Table 6 shows the comparisons of the storage overhead. Since smart contract is used to manage pseudonyms, the storage overhead of RSU in the proposed scheme is smaller.

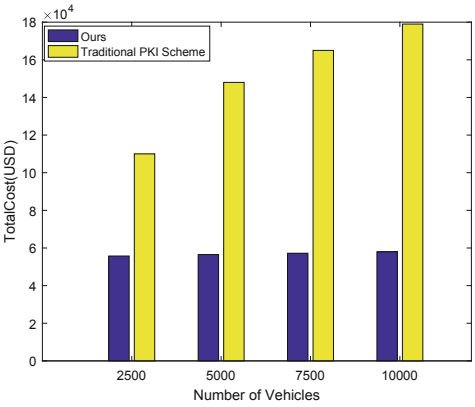

Fig. 2. Cost comparison with traditional PKI schemes

Table 6. Comparisons of storage overhead

Schemes	Standard	Ours
Vehicle	1.2 MB	2.8 MB
RSU	132 MB	30 MB

5.3 Computation Overhead

Due to the rapid change in the vehicle location and network topology, the computational overhead of RSU and OBU will affect the performance of VANET. On the one hand, RSU has a wealth of computing overhead, so we do not consider the impact of RSU's computing overhead on VANET. On the other hand, the computing power of OBU is limited, so we mainly analyzes the computing overhead of OBU. The sum of the computation overhead of V2I authentication and V2V authentication is the computation overhead of the proposed scheme.

In order to evaluate the computational cost of various cryptographic operations, the simulation platform used in the experiment is MIRACL. Each operation is performed 10,000 times on a 16 GB 64-bit Windows 10 system on an Intel(R) Core(TM) i5-1135G7@ 2.40 GHz workstation. The definition and execution time of various operations are summarized in Table 7. Compared with the

calculations summarized in Table 7, the calculation cost of hash function (T_h), point addition (T_{pa}), RSA verification (T_{R_v}) and RSA encryption (T_{R_e}) can be omitted according to [24], and according to [3], the computational overhead of RSA encryption is the same as that of RSA verification.

In the V2I authentication protocol of the proposed scheme, OBU verifies the signature of RSU and generates a signature. During the verification process, OBU needs to calculate the shared key, which is equivalent to two RSA encryption operations. Therefore, the computational cost of the proposed scheme in the V2I authentication process is $T_{bp} + 2T_{pm} + 2T_{ep2} + T_{mul} + 2T_{R_e}$.

During the V2V authentication process, all OBUs use pseudonyms for communication, so OBU first verifies whether RSU's signature on the pseudonyms is valid. Next, if the pseudonym signature is valid, OBU will perform a RSA verification operation on the signed message. OBU also needs to perform RSA encryption to generate its own signed message. Finally, the computational cost of the proposed scheme in the V2V authentication process is $T_{bp} + T_{pm} + T_{ep2} + T_{mul} + T_{R_\sigma} + T_{R_v}$.

In this section, we compare the performance of the proposed scheme with other three schemes in terms of computational overhead. The computational costs of the four schemes are evaluated and summarized in Table 8. The comparative analysis shows that in the V2I and V2V authentication process, the proposed scheme owns the lower computational cost than the other three schemes.

Table 7. The definition and execution time of related operations

Operations	Definitions	Time (ms)
T_{bp}	Bilinear pairing operation (bp)	1.6
T_{mtp}	Map-to-point hash operation (mtp)	0.8
T_{ep^2}	Exponentiation in G2 of the bilinear pairing (ep2)	0.6
T_{mul}	Scale multiplication (mul)	0.533
T_{R_σ}	RSA sign (R_σ)	0.533

Table 8. Computational cost in the V2I and V2V authentication

Schemes	V2I Authentication	V2V Authentication	Computational Time (ms)
PACE [3]	$5T_{bp} + 17T_{pm}$ $+2T_{ep2} + T_{mul}$	$5T_{bp} + 15T_{pm}$ $+2T_{ep2} + T_{mul}$	45.06
CPAS [7]	$3T_{mtp} + 3T_{bp}$ $+7T_{pm}$	$3T_{mtp} + 3T_{bp}$ $+5T_{pm}$	24
ACPN [10]	$5T_{mtp} + 5T_{bp}$ $+4T_{pm} + T_{R_e}$	$2T_{mtp} + T_{bp}$ $+3T_{pm}$	20.8
Ours	$T_{bp} + 2T_{pm} + 2T_{ep2}$ $+T_{mul} + 2T_{R_e}$	$T_{bp} + T_{pm} + T_{ep2}$ $+T_{mul} + T_{R_\sigma} + T_{R_v}$	9

6 Conclusions

In the paper, we have proposed an efficient and automatic pseudonym management scheme for VANET. By using blockchain technology to manage users' anonymous credential and pseudonym materials, the proposed scheme can reduce the cost and improve the efficiency compared to the traditional certificate-based PKI scheme for VANET. When a vehicle is driving across domains, a pseudonymous certificate can be used for cross-domain authentication at a nearby RSU, which effectively protects the privacy of the vehicle. In addition, when RSU discovers that a vehicle reports incorrect traffic information, it will trigger a smart contract to vote on the vehicle, so that the anonymity of the vehicle can be revoked more reasonably. Security and performance analysis shows that the proposed scheme is secure and practical. The future work is to present a more effective pseudonym generation and dynamic mechanism for VANET.

References

1. Albarqi, A., Alzaid, E., Al Ghamdi, F., Asiri, S., Kar, J.: Public key infrastructure: a survey. J. Inf. Secur. **6**(1), 31–37 (2015)
2. Calandriello, G., Papadimitratos, P., Hubaux, J.P., Lioy, A.: Efficient and robust pseudonymous authentication in VANET. In: Proceedings of the fourth ACM international Workshop on Vehicular Ad Hoc Networks (VANET2007), pp. 19–28, ACM (2007). https://doi.org/10.1145/1287748.1287752
3. Huang, D., Misra, S., Verma, M., Xue, G.: PACP: an efficient pseudonymous authentication-based conditional privacy protocol for VANETs. IEEE Trans. Intell. Trans. Syst. **12**(3), 736–746 (2011)
4. Intelligent Transportation Systems Committee of the IEEE Vehicular Technology Society. In: IEEE Trial-Use Standard for Wireless Access in Vehicular Environments-security Services for Applications and Management Messages, IEEE Standard, 1609.2-2006, pp. 1–105, IEEE (2006). https://doi.org/10.1109/IEEESTD.2006.243731
5. Intelligent Transportation Systems Committee of the IEEE Vehicular Technology Society. In: IEEE Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages, IEEE Standard, 1609.2-2016, pp. 1–240, IEEE (2016). https://doi.org/10.1109/IEEESTD.2016.7426684
6. European Telecommunications Standard Institute: Intelligent Transport Systems (ITS) Security; Trust and Privacy Management; Release 2 ETSI Standard, ETSI-TS 102 941, pp. 1–83, ETSI (2021). https://standards.globalspec.com/std/14474013/TS%20102%20941
7. Shim, K.A.: CPAS: an efficient conditional privacy-preserving authentication scheme for vehicular sensor networks. IEEE Trans. Veh. Technol. **61**(4), 1874–1883 (2012)
8. Lasla, N., Younis, M., Znaidi, W., Arbia, D.B.: Efficient distributed admission and revocation using blockchain for cooperative ITS. In: 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS'18), pp. 1–5, IEEE (2018). https://doi.org/10.1109/NTMS.2018.8328734
9. Lei, A., Cruickshank, H., Cao, Y., Asuquo, P., Ogah, C.P.A., Sun, Z.: Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. IEEE Internet of Things J. **4**(6), 1832–1843 (2017)

10. Li, J., Lu, H., Guizani, M.: ACPN: a novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs. IEEE Trans. Parallel Distrib. Syst. **26**(4), 938–948 (2015)
11. Lin, X., Sun, X., Ho, P.H., Shen, X.: GSIS: a secure and privacy preserving protocol for vehicular communications. IEEE Trans. Veh. Technol. **56**(6), 3442–3456 (2007)
12. Lin, C., He, D., Huang, X., Kumar, N., Choo, K.R.: BCPPA: a blockchain-based conditional privacy-preserving authentication protocol for vehicular Ad hoc networks. IEEE Trans. Intell. Trans. Syst. **22**(12), 7408–7420 (2020)
13. Lin, X., Li, X.: Achieving efficient cooperative message authentication in vehicular ad hoc networks. IEEE Trans. Veh. Technol. **62**(7), 3339–3348 (2013)
14. Lu, Z., Liu, W., Wang, W., Qu, G., Liu, Z.: A privacy-preserving trust model based on blockchain for VANETS. IEEE Access **6**, 45655–45664 (2018)
15. Ma, Z., Zhang, J., Guo, Y., Liu, Y., Liu, X., He, W.: An efficient decentralized key management mechanism for VANET with blockchain. IEEE Trans. Veh. Technol. **69**(6), 5836–5849 (2020)
16. Papadimitratos, P., et al.: Secure vehicular communication systems: design and architecture. IEEE Commun. Mag. **46**(11), 100–109 (2008)
17. Qi, J., Gao, T.: A privacy-preserving authentication and pseudonym revocation scheme for VANETs. IEEE Access **8**, 177693–177707 (2020)
18. Qu, F., Wu, Z., Wang, F.Y., Cho, W.: A security and privacy review of VANETs. IEEE Trans. Intell. Trans. Syst. **16**(6), 2985–2996 (2015)
19. Raya, M., Hubaux, J.P.: Securing vehicular ad hoc networks. J. Comput. Secur. **15**(1), 39–68 (2007)
20. Shao, J., Lin, X., Lu, R.X., Zuo, C.: A threshold anonymous authentication protocol for VANETs. IEEE Trans. Veh. Technol. **65**(3), 1711–1720 (2016)
21. Tikhomirov, S., Voskresenskaya, E., Ivanitskiy, I., Takhaviev, R., Alexandrov, Y.: Smartcheck: static analysis of Ethereum smart contracts. In: 2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB'18), pp. 9–16, IEEE/ACM (2018). https://doi.org/10.1145/3194113.3194115
22. VeriSign.: total cost of ownership for public key infrastructure, Patent (2005). http://www.imaginar.org/sites/ecommerce/index_archivos/guias/G_tco.pdf
23. Vijayakumar, P., Chang, V., Deborah, L.J., Balusamy, B., Shynu, P.G.: Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks. Future Gener. Comput. Syst. **78**, 943–955 (2018)
24. Vijay, S., Sharma, S.C.: Threshold signature cryptography scheme in wireless ad-hoc computing. Contemp. Comput. **40**(7), 327–335 (2009)
25. The costs of managed PKI: in-house implementationof PKI vs. traditional managed PKI vs. on-demand PKI, patent (2006). https://azslide.com/the-costs-ofmanaged-pki_59892e9b1723dda4299be2/36.html
26. Zhang, L., Wu, Q., Qin, B., Domingo-Ferrer, J., Liu, B.: Practical secure and privacy-preserving scheme for value-added applications in VANETs. Comput. Commun. **71**, 50–60 (2015)
27. Zhong, H., Huang, B., Cui, J., Li, J., Sha, K.: Efficient conditional privacy-preserving authentication scheme using revocation messages for VANET. In: 27th International Conference on Computer Communication and Networks (ICCCN2018), pp. 1–8. IEEE (2018). https://doi.org/10.1109/ICCCN.2018.8487337
28. Zhu, X., Jiang, S., Wang, L., Li, H.: Efficient privacy-preserving authentication for vehicular ad hoc networks. IEEE Trans. Veh. Technol. **63**(2), 907–919 (2014)