An Efficient Privacy-preserving Authentication Model Based on Blockchain for VANETs

Xia Feng[a], Qichen Shi[b], Qingqing Xie[b] and Lu Liu [c]

[a] *School of Automotive and Traffic Engineering, Jiangsu University, China*
[b] *School of Computer Science and Communication Engineering, Jiangsu University, China*
[c] *School of Informatics, University of Leicester, England*

ARTICLE INFO

Keywords:
Privacy-preserving
Authentication
Accumulator
Blockchain
VANETs.

ABSTRACT

The existing privacy-preserving authentication models for Vehicular Ad-hoc Networks (VANETs) primarily preserve multiple pseudonyms for one vehicle, while overlooking the consideration of confidential identity requirements. These authentication models cause pseudonyms management complex and revocation inconvenient. Blockchain seems to be suitable for storing the pseudonym certificates as transactions in the ledger, which enables distributed authentication. However, blockchain produces high latency for the membership verification of users. To tackle these problems, we present an Efficient Privacy-preserving Authentication Model (EPAM), leveraging the asynchronous accumulator to extend the blockchain application. The asynchronous accumulator supports efficient membership verification and avoids the time consuming of checking the Certificate Revocation List (CRL). Additionally, by designing a mutual authentication protocol, we achieve privacy properties such as anonymity and unlinkability under the consideration of the semi-trust RSUs. The simulations show that the membership verification time is about 0.157ms in EPAM test over 10^7 certificates, thus alleviating the mutual authentication latency in VANETs.

1. Introduction

Vehicular Ad-hoc Networks (VANETs) emerge as a powerful solution in the Intelligent Transportation System (ITS). VANET is a heterogeneous network that incorporates communications among vehicle and other road facilities[1]. This heterogeneous network enables vehicles to exchange time-critical information such as road safety, navigation, and other roadside services over the wireless network[2]. However, the architecture of network brings in massive real-time messages propagation and dissemination, which would be leveraged by the adversaries to perform data association, integration analysis and privacy mining. To cope with the problems, a privacy-preserving authentication scheme should be established so that vehicles can communicate with each other in a secure way.

Applying pseudonyms mechanism to protect the real identity [3][4] is a natural idea. However, there can be a powerful adversary may try to link the new and old pseudonyms by monitoring the temporal and spatial relations [5]. Moreover, these schemes require fresh pseudonym and certificate for each authentication process. With a large number of pseudonyms and certificates reserved in the On-board Units (OBUs), it will be inconvenient to conduct a revocation scheme when the malicious vehicle be found in the network.

At first glance, blockchain seems to be suitable for storing the pseudonym certificates in the ledger, as it allow multiple parties to enable a distributed authentication [6]. In particular, Yao and Chang presented an anonymous authentication scheme based on blockchain, where several fog nodes are adopted to support certificate issuance and consensus reaching [7]. However, the introduction of fog units might increase vulnerable nodes, and the risk of privacy leakage. Lu et al. in [8] proposed a privacy-preserving signature verification scheme for communication, where the distributed authentication can be processed by individual vehicles. However, with the number of vehicles increases, computation and transmission overhead of RSUs will increase linearly. Moreover, those privacy-preserving schemes [9] [10] are time-consuming processes and may fail to satisfy the efficient requirement because of the dynamic topology and high node mobility in VANETs.

Our contributions. In the scheme, our goal is to tackle the problem of the aforementioned schemes and propose an efficient privacy-preserving authentication model suitable for VANETs based on blockchain. We leverage the asynchronous accumulator presented by Leonid Reyzin and Sophia Yakoubov [11], which requires only a logarithmic update frequency. With a blockchain network storing the vehicle certificates and authentication results, the accumulator provides efficient membership verification. In this way, distributed authentication is enabled to meet the time-critical processes. We also design a mutual protocol to cope with the semi-trusted RSUs. In summary, the contributions are as follows:

- We adopt a novel membership verification model named asynchronous accumulator to extend the blockchain application, which supports the distributed authentication. By recomputing the vehicle's certificate and its witness, referring to several one-way hash calculations, the vehicle will be proved to be an authorized entity efficiently. Moreover, our new scheme can support non-membership verification and avoid the time consuming for checking Certificate Revocation List (CRL).
- We present a mutual authentication protocol under the consideration of the semi-trust model of RSUs. By randomizing the messages with Elliptic-curve cryptosystems, our scheme achieves identity anonymity as well as unlink-

✉ xiazio@ujs.edu.cn (X. Feng); ericshiqichen@gmail.com (Q. Shi); xieqq@ujs.edu.cn (Q. Xie); l.liu@leicester.ac.uk (L.L.)
ORCID(s): 0000-0003-3677-6823 (X. Feng)

ability during the authentication procedure.

- We implement EPAM in Fabric and conduct extensive simulations. The comparison of the existing blockchain-based schemes indicates that our scheme achieves efficiency of the mutual authentication.

The remainder of our paper is organized as follows: a survey of existing authentication models for VANETs is given in Section II, Section III introduces the asynchronous accumulator and describes the security issues and threat model in the current model. Section IV presents our proposal EPAM, which is an efficient privacy-preserving authentication model based on blockchain. The analysis of the security and privacy of our model is in Section V. We evaluate our model and compare it with the existing proposals in Section VI. Finally, we conclude our proposal as well as the future work in Section VII.

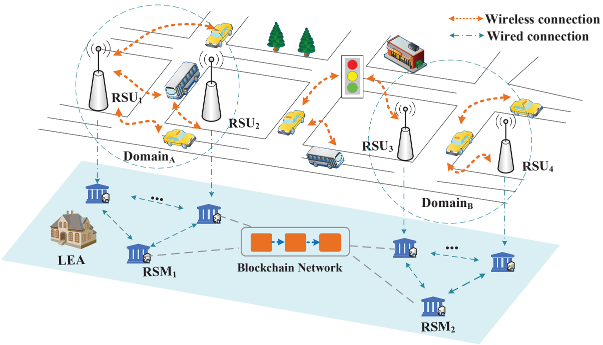


Figure 1: System model of VANETs

2. Related work

We discuss the current work related to our model in this section. Particularly, we cover blockchain-based authentication and privacy-preserving schemes. The articles we mentioned below make use of more than one technology to construct their architectures. For the convenience of illustration, we classify the articles according to their domain methodology.

2.1. Blockchain-based approaches

Compared with the conventional schemes, decentralized authentication schemes are more suitable for VANETs. The core idea of decentralized schemes is to eliminate the single point of failure and support the distributed network. Approaches to building a more reliable authentication have sought to apply blockchain technology to decrease the role of certificates or CAs. Certcoin [12] is a PKI architecture supporting decentralized authentication, which is based on namecoin [13]. It stores identity and public key in a Merkle hash tree accumulator and leverages distributed hash table to support efficient retrieval. Axon [14] introduced a privacy-awareness mechanism to implement a more reliable PKI. It aims to eliminate the public link between public keys and identity to avoid adversaries tracking actions. It is mentioned [14] that the privacy-awareness architecture can achieve either neighbor group anonymity or totally anonymity. However, the author did not provide a concrete scheme to implement the architecture.

2.2. Privacy-preserving approaches based on blockchain

In order to preserve privacy with the concept of applying blockchain to authenticate vehicles in VANETs. Various authentication mechanisms have been presented. BPPA [8] applied a Merkle Patricia tree (MPT) to provide a distributed authentication scheme without the revocation list. It achieves conditional privacy and allows a vehicle to use multiple certificates. The certificates and real identity are encrypted and stored in the blockchain. The linkability between the certificates and real identity can only be revealed in case of disputes. BPPA conducts an experiment and shows that its authentication can be processed by individual vehicles within 1 ms. However, the experiment didn't consider the communication delay between vehicles and RSUs. BLA [7] proposed a blockchain-based authentication scheme assisted by fog data-centers, achieving anonymity, and granting vehicle users the responsibility of preserving their privacy. All the entities in BLA are assumed trustworthy, which is hard to satisfy in the realistic world. BUA [15] proposed a blockchain-based unlinkable authentication protocol, vehicles use homomorphic encryption to generate pseudonyms to achieve unlinkability. However, the communication delay for vehicles in BUA's simulation is measured in seconds, which is hard to satisfy the mobility of VANETS.

However, such methodologies have several drawbacks. Firstly, they need extra computing of vehicles to generate pseudonyms or encrypt real identity to achieve unlinkability, which incurs additional computation and time cost. Secondly, during the authentication process, vehicles need to query a certificate revocation list (CRL) to check the current status of the certificate in some of these schemes. Thirdly, the problem of low efficiency in some mechanisms hasn't been resolved. To this end, we propose a novel EPAM (Efficient Privacy-preserving Authentication Model) to facilitate the authentication in a decentralized way.

3. Prelimiaries

Before presenting our scheme, we introduce a brief review of the preliminary knowledge. System and security model will also be presented in this section.

3.1. Asynchronous accumulator

In our scheme, we extended blockchain application by constructing asynchronous accumulators. The accumulators are used to reduce the membership verification latency. Membership verification is defined as the procedure for each vehicle that has been registered to LEA can be verified as a member of VANETs. We consider the membership verification as an important procedure for our authentication model.

The accumulator was firstly proposed by Benalon and De Mare [16] in 1993. An accumulator is a set of elements organized in a more compact way, which is used to retrieve and verify membership efficiently. Leonid Reyzin and Sophia Yakoubov [11] introduced an efficient asynchronous accumulator suitable for distributed applications and peer to peer network. It leverages Merkle hash trees, but maintains multiple Merkle tree roots as part of the accumulator value [17]. This kind of accumulator is suitable for VANETs, because it requires only a logarithmic update frequency and supports the verification of an up-to-date witness against an outdated accumulator.

In our construction of asynchronous accumulator, the procedure is similar to build a Merkle hash tree. As shown in Figure 2, the shaded part is an accumulator. The leaf nodes are $H(Cre_{v_0})$,$H(Cre_{v_1})$,..., $H(Cre_{v_n})$, in which $H(Cre_{v_i})$ is the vehicle certificate operated by a one-way hash function. The elements with blue dashed outlines are the witness for element $H(Cre_{v_1})$, which is marked with red line. Witness for $H(Cre_{v_1})$ is $w_{(Cre_{v_1})} = ((H(Cre_{v_0}), left)), (H(H(Cre_{v_2})\| H(Cre_{v_3})), right)), H(H(Cre_{v_4})\|H(Cre_{v_7)}), right))$.

For constructing the second-last level of the tree, starting from left, a node is inserted at the level acts as a parent for every two leaf nodes, that would be $Z = H(H(Cre_{v_i})\|H(Cre_{v_{i+1}}))$. This method of constructing the next higher-level node is repeated until the root is constructed. r^d is the root of a complete Merkle tree with 2^d leaves if and only if the d^{th} least significant bit of the binary expansion of n is 1. Otherwise, $r^d = \perp$. When the newly registered vehicle needs to be added to the accumulator, we should merge the Merkle trees to create deeper ones.

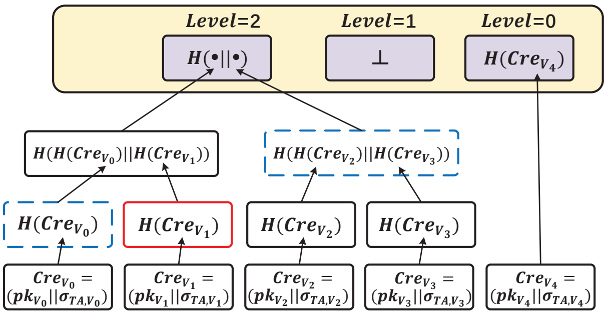

Level=2
Level=1
Level=0
$H(\bullet||\bullet)$
$\perp$
$H(Cre_{V_4})$
$H(H(Cre_{V_0})||H(Cre_{V_1}))$
$H(H(Cre_{V_2})||H(Cre_{V_3}))$
$H(Cre_{V_0})$
$H(Cre_{V_1})$
$H(Cre_{V_2})$
$H(Cre_{V_3})$
$Cre_{V_0} = (pk_{V_0}||\sigma_{TA,V_0})$
$Cre_{V_1} = (pk_{V_1}||\sigma_{TA,V_1})$
$Cre_{V_2} = (pk_{V_2}||\sigma_{TA,V_2})$
$Cre_{V_3} = (pk_{V_3}||\sigma_{TA,V_3})$
$Cre_{V_4} = (pk_{V_4}||\sigma_{TA,V_4})$

Figure 2: An illustration of accumulator.

3.2. System Model

We intend to provide an efficient privacy-preserving authentication scheme in VANETs. Figure 1 describes the system model of our proposal, which consists of four entities. There are the Law Enforcement Authority (LEA), Regional Service Managers (RSMs), Road-side Units (RSUs) and users(i.e., On-board Units). The wired connections among LEAs, RSMs and RSUs are considered reliable, while the communication between RSUs and vehicles using wireless connections is vulnerable. A blockchain network is constructed to store the vehicle certificates and authentication results permanently and immutably. We introduce the system model with the main components.

Law Enforcement Authority (LEA): LEA should be an institution authorized by law that is a trusted management center. It provides authorization to the vehicles and RSMs in our model. LEA has the ability to audit the network by revealing the real identities of the RSUs and vehicles.

Regional Service Manager (RSM): RSM roles that are delegated by the LEA to assure valid and efficient authentication. RSMs are fully-trusted and divide the whole network into several logical domains. Each RSM is responsible for registering and revoking vehicles as well as RSUs within its domain. They also act as blockchain and accumulator managers, responsible for maintaining the public Leger and constructing the asynchronous accumulator. Moreover, they are responsible for constructing a key-value pair table, for example, a Distributed Hash Table (DHT) [18] to preserve the vehicle certificates and the corresponding witnesses.

Road-side Units (RSUs): RSUs are located alongside the roads to organize and coordinate vehicular communications in an optimized manner. Considering the development of hardware, we assume that the RSUs have stronger computation and storage capabilities They are semi-trusted [19] facilities, in the sense that they may misbehavior on their own but would never conspire with either of the other entities.

On-board Units(OBUs): The main computing and communication units in vehicles are the OBUs. They are embedded equipment with limited computing capability, which can be used to communicate with each other.

3.3. Security model

In [20], Raya et al. divided the applications into two major categories: safety-related and value-added applications. Based on the categories, [4] defined four basic security threats including bogus information, identity disclosure, denial of services, replay attacks. Thus, the following security and privacy-preserving requirements should be satisfied:

- **Authentication correctness and security:** For the correctness property, the authorized vehicles can always be verified that they are indeed the legal entities. And referring to security, EPAM can be proved that it can resist the basic attacks as well as replay attack[21]. After the authentication, a server can establish a secure channel between vehicles or RSUs, and no adversary can tamper with transmitted messages.

- **Replay attack resistance:** Our scheme can resist the attacker who repeats a previously transmitted message, with the intention of intercepting and retransmitting its modified version, thereby fooling the honest authentication party.

- **Nonrepudiation:** LEA has the ability to trace the misbehaving vehicles and reveal their real identities.

- **Privacy preservation:** In our scheme, the privacy-preserving property refers to 1) anonymity, the ability to communicate without revealing the identity of the vehicles, and 2) unlinkability, the ability of a single vehicle to send multiple messages without revealing that the messages are sent by the same vehicle.

4. EPAM (Efficient Privacy-preserving Authentication Model)

The section presents EPAM, including three phases: initialization, chain construction, and mutual authentication, as shown in Figure 3. LEA, RSM, RSU and vehicles are initialized in Phase I, while the blockchain and accumulators are generated in Phase II. The mutual authentication protocol is conducted in

Table 1 Basic Notations and Description

Notation	Description
R_i	The i^{th} RSU.
RSM_i	The i^{th} RSM.
pk_{LEA}, sk_{LEA}	Public key and secret key of LEA.
pk_i^{RSM}, sk_i^{RSM}	Public key and secret key of RSM$_i$.
pk_i^R, sk_i^R	Public key and secret key of R_i.
pk_i^V, sk_i^V	Public key and secret key of v_i.
σ_{RSM,R_i}	Signature signed by LEA's secret key for R_i.
σ_{RSM,V_i}	Signature signed by LEA's secret key for v_i.
$\|\|$	Message concatenation.
T_R	A timestamp of RSU.
Cre_{v_i}	Certificate of v_i generated by LEA.
$\{M\}_{pk}$	Encrypt M with public key pk.

phase III. EPAM segregates the network into two private subset communication channels, which would conduct private and confidential transactions. The channel between LEAs and RSMs called Channel A shares the vehicle's certificate information. While the channel between RSMs and RSUs called Channel B shares the authentication results. Besides, we provide revocation mechanism for the vehicles. The frequently used notations and parameters are described in Table 1.

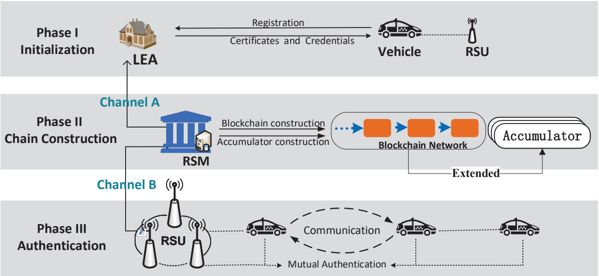


Figure 3: The architecture of EPAM

4.1. System Initialization

The initialization phase is conducted by the LEA. We employ the Elliptic-curve cryptosystems as the underlying primitives for RSMs, RSUs and OBUs.

1) LEA's initialization

- Set an elliptic curve $E : y^2 = x^3 + Ax + B$, where $A, B \in \mathbb{Z}_p^*$ are constants with $4A^3 + 27B^2 \neq 0 \bmod p$
- Let $p \geq 5$ be a prime, $\mathbb{E}_q(a, b)$ be the elliptic group of points, q is the order and G is the base point.
- LEA randomly chooses an integer sk_{LEA} from $\{1, ..., q - 1\}$, then computes its public key $pk_{LEA} = sk_{LEA} \times G$.
- A one-way hash function $h : (0, 1)^* \rightarrow (0, 1)^l$.
- LEA publishes the public parameters $(A, B, G, p, q,$ pk_{LEA}) and reserves its secret key.

2) RSM's initialization

- Each RSM_i randomly chooses an integer sk_i^{RSM}, then computes its public key $pk_i^{RSM} = sk_i^{RSM} \times G$.
- All the RSMs deliver their secret and public key pairs to the LEA via a secure channel.
- Each RSM broadcasts its public key in its domain.

3) RSU's registration

For the registration of R_i , LEA verifies the RSU's identity and issues the public key for R_i.

- LEA chooses a random number sk_i^Rwhich is the private key of R_i, then computes the public key $pk_i^R = sk_i^R \times G$.
- LEA signs RSU with its secret key, $\sigma_{LEA,R_i} = Sig(pk_i^R \|$ $D_A, sk_{LEA})$. The certificate for R_i is $(pk_i^R,\sigma_{LEA,R_i})$.
- LEA delivers sk_i^R, pk_{LEA} and its certificate via a secure channel to R_i.

4) Vehicle's registration

After verifying the identity of vehicle thoroughly, RSM issues certificates for vehicle as follows:

- LEA generates a secret integer sk_i^v as the private key for v_i, then computes the public key $pk_i^v = sk_i^v \times G$.
- LEA signs v_i with its secret key, $\sigma_{LEA,V_i} = Sig(pk_i^v,$ $sk_{LEA})$.
- Vehicle reserves its certificate, $Cre_{v_i} = (pk_i^v \| \sigma_{LEA,v_i})$in the OBU, which was delivered by LEA via a secure channel.

In this initialization procedure, LEA should retain the vehicle's private material and its offline secret key confidentiality. Thus, we assume that there is no privacy disclosure and security attack risk in this phase.

4.2. Chain construction

1) Certificate transaction generation

After the initialization phase,the certificate transactions will be generated by RSM, as described below. We choose Kafka[22] as the consensus service in our model, which can handle hardware crash problems.

Step-1: The submitter RSM_s initiates a transaction proposal, which is a request to invoke a vehicle's certificate transaction with input parameters $H(Cre_{v_i}) = H(pk_i^V \| \sigma_{LEA,v_i})$. Then, RSM_s packages the transaction proposal into the properly format and signs the transaction proposal with its secret key, shown as formula (1)

$$\sigma_{RSM_s,v_i} = Sig(H(Cre), sk_{RSM_s}) \tag{1}$$

Step-2: The endorsing RSMs verify 1) the format of transaction proposal is correct, 2) the transaction proposal has not been submitted in the past, 3) the signature of submitter RSM_s is valid, and 4) the submitter RSM_s is properly authorized to perform the proposed operation on the channel. After that, the endorsing RSMs take the transaction proposal inputs and sign them as following formula

$$\sigma_{RSM_k,v_i} = Sig(\sigma_{RSM_s,v_i}, sk_{RSM_k}) \tag{2}$$

Table 2 Description for Accumulator

Algorithm	Description
A_t	The accumulator at time t.
$w_t^{v_i}$	The witness for v_i in accumulator at time t.
$H(Cre_{v_i})$	The i^{th} element $H(Cre_{v_i})$ for v_i in the accumulator.
$Gen(1^k) \rightarrow A_0$	Initialize the accumulator A_0.
$Add(A_{t-1}, H(Cre_{v_i})) \rightarrow (A_t, w_t^{v_i}, M_t^{v_i})$	Update the accumulator A_{t-1} by adding element $H(Cre_{v_i})$, and returns the new state of accumulator A_t as well as the new witness $w_t^{v_i}$. Messages $M_t^{v_i}$ will be sent to witness holders to update the witness.
$WitAdd(v_i, w_{t-1}^{v_i}), M_t^{v_i}) \rightarrow w_t^{v_i}$	On receiving the message $M_t^{v_i}$ after the new element $H(Cre_{v_j})$ inserted, RSM updates the current state of witness $w_{t-1}^{v_i}$ and returns an updated $w_t^{v_i}$.
$VerMem(A_t, H(Cre_{v_i}), w_t^{v_i}) \rightarrow \{1, 0\}$	RSM checks the membership of $H(Cre_{v_i})$ with the witness and returns 1 if the certificate of v_i appears to be in A_t, otherwise returns 0.

Step-3: The submitter RSM_s verifies endorsing peers' signatures and compares the endorsement policy. Then, RSM_s submits the transaction to the ordering service to update the ledger.

Step-4: After the blocks of transactions created, a message will be sent to each RSM to notify that the transaction has been immutably appended to the chain. Upon receiving the message, each RSM updates its Ledger to the current world state.

2) Accumulator generation

After the vehicle certificates have been appended to the blockchain, each RSM constructs an asynchronous accumulator to support the efficient decentralized authentication. The whole point of the accumulator generation phase is trying to reduce the membership verification latency. In the paper, membership verification has been formalized in Formula (8) to verify whether a vehicle is a member of VANETs or not. The procedure explains as follows, and Table 2 lists the algorithms used in this phase.

Step-1: RSM initiates the Merkle hash accumulator A_0 by conducting formula (3), which representing an empty set

$$Gen(1^k) \rightarrow A_0. \tag{3}$$

Step-2: RSM adds the new vehicle's certificate $H(Cre_{v_i})$ to the accumulator, generating the updated accumulator which is A_t, and the membership witness $w_t^{v_i}$ for the vehicle v_i generates as follows

$$Add(A_{t-1}, H(Cre_{v_i})) \rightarrow (A_t, w_t^{v_i}, M_t^{v_i}). \tag{4}$$

The RSM constructs a key-value pair table, for example, a Distributed Hash Table (DHT) [18] to reserve the $H(Cre_{v_i})$ and the corresponding witness $w_t^{v_i}$.

Step-3: If the accumulator A_{t-1} already has any other elements in it, for example, $\{H(Cre_{v_1}), H(Cre_{v_2}), \ldots, H(Cre_{v_n})\}$. When a new element $H(Cre_{v_j})$ should be added in the accumulator, a subset of the update messages $\{M_t^{v_1}, M_t^{v_2}, \ldots, M_t^{v_n}\}$ will be generated and delivered to the other RSMs. The update message could be used to update the witness lists. The update message $M_t^{v_n}$ contains the element $\{w_t^{v_n}, A_{t-1}, A_t, H(Cre_{v_n})\}$.

Step-4: Once the RSMs receive the update message $M_t^{v_1}$ at time t, they can choose to update the vehicle's witness by replacing the old one, the algorithm shown as formula (5)

$$WitUpdate(H(Cre_{v_i}), w_{t-1}^{v_i}, M_t^{v_1}) \rightarrow w_t^{v_i}. \tag{5}$$

After the accumulator generation phase, each RSM has the up-to-date accumulator and the corresponding key-value pair table. Note that, the accumulator [11] used in our scheme has low update frequency and old accumulator compatibility, which can support authentications in VANETs in a better way.

4.3. Certificate revocation

In our scheme, LEA could revoke a misbehaving vehicle before the expiration time of its certificate, the revocation steps are listed as follows:

Step-1: LEA sends a transaction proposal to RSM, which is a request to revoke a vehicle's certificate transaction with input parameters $(revoke, H(pk_{LEA} \| \sigma_{LEA, v_i}))$. RSM updates the ledger with a revocation transaction. The transaction commitment procedure is almost the same as described in Section 4.B-1 from Step 2 to Step 4.

Step-2: RSM updates the accumulator by replacing the leaf node $H(Cre_{v_i})$ with $\perp$. Then, RSM broadcasts the update message.

Note that the revocation check time will be significantly decreased by the using of asynchronous accumulator mechanism.

4.4. Mutual authentication

In this phase, we introduce an authentication protocol shown as Figure 5, including five steps described as follows. The authentication results will be saved in the blockchain by the RSM. It is enabled by the progress described below.

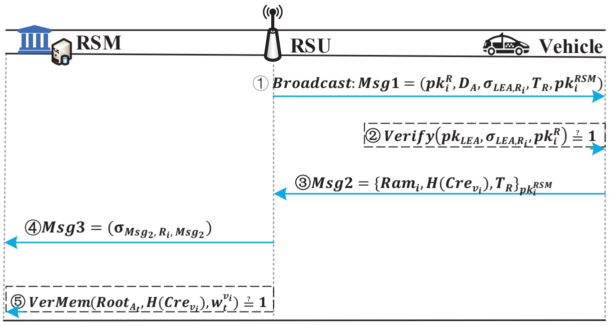

RSM
RSU
Vehicle
① $Broadcast: Msg1 = (pk_i^R, D_A, \sigma_{LEA,R_i}, T_R, pk_i^{RSM})$
② $Verify(pk_{LEA}, \sigma_{LEA,R_i}, pk_i^R) \overset{?}{=} 1$
③ $Msg2 = \{Ram_i, H(Cre_{v_i}), T_R\}_{pk_i^{RSM}}$
④ $Msg3 = (\sigma_{Msg_2, R_i, Msg_2})$
⑤ $VerMem(Root_{A_t}, H(Cre_{v_i}), w_t^{v_i}) \overset{?}{=} 1$

Figure 4: Mutual authentication protocol of EPAM

Step-1: RSU periodically broadcasts its public key, domain, timestamp, communication rage and speed limit. For RSU R_i in

Domain D_A, the message is $Msg_1 = (pk_i^R, D_A, \sigma_{LEA,R_i}, T_R, pk_i^{RSM})$.

Step-2: On receiving the message, the vehicle firstly checks whether D_A is a new domain. If it stands, the vehicle v_i initiates the verification procedure. By running formula (6)

$$Verify(pk_{LEA}, \sigma_{LEA,R_i}, pk_i^R) \overset{?}{=} 1. \tag{6}$$

v_i could check the validation of R_i.

Step-3: If R_i is valid, v_i sends $Msg_2 = \{Ram_i, H(Cre_{v_i}), T_R\}_{pk_i^{RSM}}$ to R_i, the message encloses hashed certificate $H(Cre_{v_i})$, a random number Ram_i and T_R, which is encrypted by the public key of RSM_i.

Step-4: R_i receives the message and signs the message with its secret key, shown as formula (7)

$$\sigma_{Msg_2,R_i} = Sig(Msg_2, sk_i^R). \tag{7}$$

then sends $Msg_3 = (\sigma_{Msg_2,R_i}, Msg_2)$ to RSM_i within its domain.

Step-5: On receiving Msg_3 at T^*, the RSM_i validates the signature of R_i and decrypts the message with the secret key. By checking whether $T^* - T_R \le \Delta T$, RSM_i ensures that the message is within the timestamp limit. Then, the RSM verifies the membership of v_i by recomputing the root of the accumulator $Root_{A_t}$ using $H(Cre_{v_i})$ and the corresponding witness $w_t^{v_i}$, as shown in formula (8)

$$VerMem(Root_{A_t}, H(Cre_{v_i}), w_t^{v_i}) \overset{?}{=} 1. \tag{8}$$

If it stands, RSM_i believes that v_i is indeed the authorized vehicle. Then RSM_i sends the result to R_i. R_i believes that v_i is the authorized vehicle and then negotiates the session key with v_i.

The authentication phase involves three entities to cope with the semi-trusted RSUs. In the reality situation, wired connection between RSMs and RSUs ensures reliable connections. Thus, the communication overhead is dominant to the wireless connection. Moreover, we achieve levels of privacy during this phase, the analysis will be provided in Section V.

5. Security and privacy analysis

We assume that the adversary in our model cannot break the standard cryptographic primitives, for instance finding hash collisions on SHA3 or forging digital signatures on Elliptic-curve cryptosystems. Further, the adversary cannot compromise the secret keys of RSUs and vehicles. Under the assumption, we can describe the security and privacy-preserving properties that EPAM is expected to have. Formally, the correctness property requires that for each vehicle who has registered it should be easily authenticated by any RSM in VANETs, and the security property requires that for each vehicle who hasn't registered it should be infeasible to prove the membership.

5.1. Authentication correctness

The proposed scheme is correct if an authorized vehicle can always be authenticated by RSM. More formally, for each vehicle v_i who has the corresponding Cre_{v_i} issued by LEA, the following holds true:

$$VerMem(Root_{A_t}, H(Cre_{v_i}), w_t^{v_i}) = \begin{cases} 1, & if \quad Cre_{v_i} \in A_t, \\ 0, & if \quad Cre_{v_i} \notin A_t, \\ \perp, \quad if \quad Cre_{v_i} \in revoked. \end{cases} \tag{9}$$

5.2. Authentication security

Our scheme achieves the security of authentications between RSUs and vehicles if 1) it is hard to fabricate a certificate Cre_{v_i} for vehicle v_i that has not been registered to the LEA, 2) it can resist replay attack, and 3) nonrepudiation.

Challenge 1: An adversary $\widetilde{A}$ pretends to be an authorized vehicle. It may forge a certificate and try to fool the nearby RSUs into believing that it is a legal entity.

Resistance: To be authenticated by RSMs, v_i performs a one-way hash function to generate its certificate hash $H(Cre_{v_i})$ based on the certificate which has been reserved in a tamper-proof device. Without knowing Cre_{v_i}, it is computationally infeasible for adversary $\widetilde{A}$ to forge a valid $H(Cre_{v_i})$, similarly, no adversary can forge a witness corresponding to $H(Cre_{v_i})$.

Challenge 2: An adversary $\widetilde{A}$ pretends to be an authorized RSU. It may forge a certificate and tries to cheat the vehicle into believing that it is a legal RSU. The communication between the authorized vehicle and the adversary $\widetilde{A}$ may disclose vehicle's identity information.

Resistance: In our protocol, when vehicle v_i receives the broadcasting message $\{pk_i^{\widetilde{A}}, D_A, \sigma_{LEA,\widetilde{A}}, T_{(\widetilde{A})}\}$ sent by $\widetilde{A}$, v_i validates the $\widetilde{A}$'s certificate by running formula (10)

$$Verify(pk_{LEA}, \sigma_{LEA,\widetilde{A}}, pk_i^{\widetilde{A}}) \overset{?}{=} 1. \tag{10}$$

Because $\widetilde{A}$ hasn't been authorized by the RSM, the verification process fails. The message sent by $\widetilde{A}$ will be dropped. The authorized vehicle wouldn't send any message to the adversary $\widetilde{A}$.

Resisting Relay attack: During the authentication phase, the message's freshness is guaranteed by ΔT. Set T_R as the moment the RSU broadcasts the message, T^* as the moment the RSM receiving the message replied from the vehicle. When receiving the message, RSM_i first checks the $T^* - T_R \le \Delta T$ to make sure the message within the timestamp limit.

Nonrepudiation: The vehicle's real certificate Cre_{v_i} is enclosed in the authentication message $Msg2$. Though the certificate hash $H(Cre_{v_i})$ and combination of encryption by RSM's secret key, no entity except for RSM can reveal the real certificate of the vehicle. If RSM suspects a vehicle, it can decrypt $Msg2$ and verify the message. Thus the nonrepudiation of our scheme is guaranteed.

5.3. Privacy analysis

It is worthy of note that our scheme predominantly intends to preserve anonymity and unlinkability as described in Section

3.3.

Anonymity: First, the real identities are reserved in the blockchain network. EPAM segregates the network into channels, where the channel represents only RSMs and LEAs can visit the data. Second, EPAM can guarantee the confidentiality of the messages during the authentication phase. Adversaries can get the real identity of vehicles because 1) identity related certificate has been hashed, 2) the message is encrypted by the public key of fully-trusted RSM, and 3) the random number and timestamp are added to make sure the authentication message is unique.

Unlinkability: We present a model to illustrate the challenges and our scheme achieves unlinkability during the authentication. We subject our scheme to two challenges: 1) the amount of identity information obtained by the attacker during the interactive authentication process, and 2) the total number of interactions collected to deduce different authentication process to a specific vehicle.

Privacy Challenge: We assume that adversary ($\widetilde{A}$) who is a semi-trusted verifier, which means $\widetilde{A}$ tries to obtain the identity information of v_i. It is possible that the $\widetilde{A}$ establishes a link lead to vehicle's privacy disclosure. We present the analysis below.

Definition: Let $(v_i, \widetilde{A})$ be the parties involved in the authentication process. These parties share the input $(v_i, \widetilde{A})$ (Msg_2), which is

$$\chi = \{Ram_i, H(Cre_{v_i}), T_{RSU}\}_{pk_i^{RSM}}. \tag{11}$$

For input Msg_2, we say that v_i and $\widetilde{A}$ use a privacy-preserving protocol has unlinkability property if the following is true.

The adversary $\widetilde{A}$ cannot determine whether Msg_2 and $Msg_2^{'}$ originated from the same vehicle or not. For Step-3 involved in the protocol, v_i sends the message $Msg_2 = \{Ram_i, H(Cre_{v_i}), T_{RSU}\}_{pk_i^{RSM}}$ to R_i, the message encloses hashed certificate $H(Cre_{v_i})$, a random number Ram_i and T_{RSU}, which is encrypted by the public key of RSM_i. With Ram_i and $T_{\widetilde{A}}$ randomize Msg_2 sent by vehicle every time, even if $\widetilde{A}$ rejects all the validations and obtains the reduplicate interactive information, it still cannot obtain any additional information to deduce different messages linking to a specific vehicle.

6. Performance evaluation

In this section, We provide the costs of running Kafka consensus in the prototype. The authentication overhead and time consumption for mutual authentication in VANETs are analyzed. Moreover, the practical viability of our proposed scheme is evaluated against the existing state-of-the-art approaches, in order to exhibit the efficiency of our proposed scheme.

6.1. Experiment setting

The prototype is implemented in Hyperledger Fabric, its chaincodes are developed in Go, a fast and compiled language. To measure the authentication overhead, we leverage the accumulator and developed it in JAVA. The ns-3 [23] is used to simulate the average communication latency between RSUs and vehicles. The protocol is IEEE 802.11p and the routing is AODV [24].

Table 3 Comparison for computation overhead

Scheme	Authenticate a vehicle	Authenticate n vehicles
IBV	$3T_{bp} + T_{pm_{ec}} + T_{mtp}$	$3T_{bp} + nT_{pm_{ec}} + nT_{mtp}$
ABAKA	$3T_{pm_{ec}}$	$(2n+1)T_{pm_{ec}}$
CPAS	$3T_{bp} + T_{sm}$	$3T_{bp} + (n+1)T_{sm}$
BPPA	$2T_{pm_{ec}} + T_{pa_{ec}} + 25T_h$	$2nT_{pm_{ec}} + nT_{pa_{ec}} + 25nT_h$
Ours	$8T_{pm_{ec}} + 4T_{pa_{ec}}$	$(n+7)T_{pm_{ec}} + (n+3)T_{pa_{ec}}$

We have three organizations, which are LEAs, RSMs and RSUs. RSMs are responsible for proposing the certificate transactions and committing them. After a certificate transaction is successfully committed, the submitter RSM broadcasts the message to RSMs and LEAs. RSMs are used as endorsing peers, while keep the certificate ledgers sync with LEA and construct the asynchronous accumulator. The endorsement policy is set that a transaction will be successfully committed when more than $2n + 1$ signatures from the endorsing RSMs are valid.

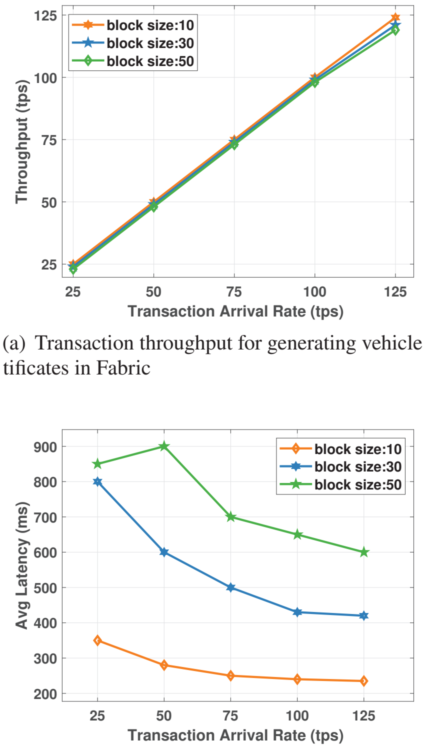
We set two private subset communication channels among LEAs, RSMs and RSUs, which would conduct private and confidential transactions. The channel between LEAs and RSMs called Channel A shares the vehicle's certificate information. While, the channel between RSMs and RSUs called Channel B shares the authentication results. The channel's policy is set that LEAs and RSMs have the right to write in Channel A and Channel B separately. Under the consideration of vehicle's mobility and limited computing power, vehicles in our scheme have no rights to access the channels.

6.2. Overhead of blockchain construction

We ran the prototype to evaluate the overhead of blockchain construction, which is measured by throughput and latency of transactions. Based on [25], throughput is defined as the rate at which transactions are committed to certificate ledge. Latency is the time taken from a RSM sending the transaction proposal to when the transaction commitments successfully. Throughput and latency are reported as the average measured during the steady state of our experiments. In our model, revoking a vehicle is conducted by appending a new transaction with revocation data to the blockchain. Therefore, the simulation of blockchain construction includes two kinds of transactions, which are registration and revocation. The transaction size has been determined in advance which is 4.5kb; thus, the block size is varying from containing 10 to 50 transactions. We set transaction arrival rates as 25tps and 125tps for write operation. Results are depicted in Figure 5. We can observe that throughput improves greatly with the transaction arrival rate increase from Figure 5(a). To measure the average latency, we adjust the block size from containing 10, 30, and 50 transactions separately. As Figure 5(b) shows, the average latency decreases with the transaction arrival rate increasing. The average latency of 600 ms for writing is observed when we set the transaction arrival rate as 30tps. The time is acceptable because our work performs certificate transaction generation algorithm during registration phase instead of authentication process.

Table 4 Comparison of message size

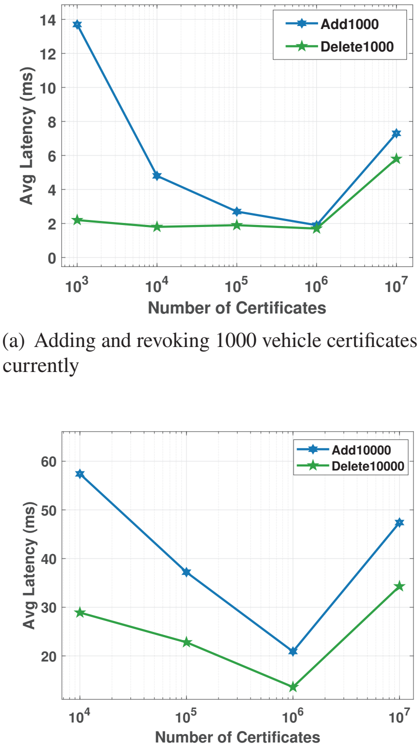
Scheme	A single vehicle		n vehicles	
	vehicle → RSU	RSU → vehicle	vehicle → RSU	RSU → vehicle
CAPS	174 bytes	143 bytes	174n bytes	143n bytes
IBV	63 bytes	N/A	63n bytes	N/A
ABAKA	63 bytes	80 bytes	63n bytes	80 bytes
Our scheme	348 bytes	85 bytes	348n bytes	85 bytes



(a) Transaction throughput for generating vehicle certificates in Fabric

(b) Latency for generating vehicle certificates in Fabric

Figure 5: Performance results via Hyberledger Fabric simulations for certificate blockchain generation in terms of the relationship between transaction arrival rate and average latency.



(a) Adding and revoking 1000 vehicle certificates concurrently

(b) Adding and revoking 10000 vehicle certificates concurrently

Figure 6: Performance results via the asynchronous accumulator simulations for vehicle certificates addition and revocation in terms of the relationship between the latency and the number of certificates.

6.3. Overhead of accumulator construction

Asynchronous accumulator is implemented to improve the efficiency of authentication. RSMs are responsible for the construction of the accumulators and reserve the witnesses according to the vehicle certificates. We run the experiments to explain the improvement in efficiency. As shown in Figure 6(a) and (b), we construct the accumulators that consist of N vehicle certificates. With the N of 10^4, 10^5, 10^6, and 10^7, we observe that the minimum average latency for adding 1000 certificates in the accumulator is 1.9 ms at the 10^6 scale. Using the same parameters, the minimum average latency for deleting is 1.7 ms. As an extended structure of blockchain, the latency for the construction of the accumulator is acceptable.

6.4. Overhead of mutual authentication

We evaluate the authentication performance, including computation cost and communication overhead analysis separately. By comparing the results with some related schemes, it is shown that our scheme achieves higher efficiency in the mutual authentication.

6.4.1. *Computation overhead analysis*

The authentication phase described in section IV has five steps. Vehicle who joins in the authentication is responsible for encrypting and sending the request message. RSU who receives the message will sign the message and forward to the RSM. Thus, the main operations in the authentication are elliptic curve point multiplication $T_{pm_{ec}}$, elliptic curve point addition $T_{pa_{ec}}$, and hash function T_h. To compare with the other schemes, we list the computation overhead for the binary paring parameters. We use the MIRACL CORE cryptographic library [26]. To evaluate the average execution time of binary paring operations, we choose a BLS12381 curve with an embedding degree 12 at a 128-bit security level. To get the average execution time of elliptic curve operations, we choose C25519 curve at a 128-bit security level. The performances ran on a Intel i5-7400 @ 3.0GHz with Ubuntu 16.04.

The computation overhead of EPAM is dominant to the number of elliptic curve point multiplication. Because in curve C25519, the time for performing $T_{pm_{ec}}$ is 0.17ms comparing with 3.5×10^{-4} ms taken by $T_{pa_{ec}}$. While the time for performing a SHA256 hash function is negligible because its execution time is only 0.0001ms. Moreover, in BLS12381 curve, the time for performing the following parameters, i.e., T_{bp}, T_{sm} and T_{mtp} are 2.644, 0.326 and 0.056 ms, respectively. We listed the performance time below.

① $T_{pm_{ec}}$ is the time required for performing an elliptic curve point multiplication, which is approximately 0.17 ms.

② $T_{pa_{ec}}$ is the time required for performing an elliptic curve point addition, which is approximately 3.5×10^{-4} ms.

③ T_{bp} is the time required for performing a bilinear pairing, which is approximately 2.644 ms.

④ T_{sm} is the time required for performing a scalar multiplication,which is approximately 0.326 ms.

⑤ T_{mtp} is the time required for perform a MapToPoint hash operation of the bilinear pairing, which is approximately 0.056 ms.

Table 3 gives the detailed comparison of computation cost among approaches including IBV [27], ABAKA[28], CPAS[29], BPPA [8]. IBV and CPAS use the bilinear pairing scheme, while ABAKA, BPPA and our scheme use the Elliptic-curve cryptosystem. The total computation overhead in terms of authentication n vehicles is illustrated in Figure 7. We observe from Figure 7 that the computation overhead linearly increases with an increasing number of vehicles. The computation overhead of BBPA is largest among these schemes, and our scheme performs better comparing with the others. When the number of vehicles increases up to 2400, our scheme saves 413 ms compared with CPAS. Note that the computation overhead of our scheme mostly relies on the RSUs and RSMs, which means vehicles with limited computation ability can cope with the overhead even for heavy traffic scenarios.

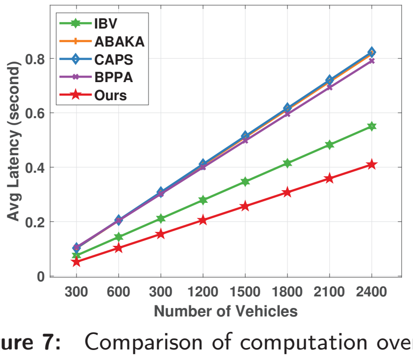


Figure 7: Comparison of computation overhead

6.4.2. *Membership verification overhead*

Before discussing the communication overhead, we implement some details for vehicle's membership verification, which is corresponding to the Step-5 of the authentication protocol in Section IV. As a matter of fact, the Fabric itself can conduct the membership verification operation. The accumulator is implemented to speed up the verification operation. The leaf nodes in accumulator is based on the world state ledger of Fabric. Thus, to prove the efficiency the accumulator achieves, we simulate experiments to measure the average latency by using Fabric and accumulator, separately.

Figure 8 (a) shows that the average latency for verifying one vehicle in Fabric is about 55ms and approaches to a constant, which is only slightly affected by the change of number of certificates.When verifying 1000 vehicles concurrently, the average latency increases with the increasing number of certificates shown in Figure 8 (b) .

Figure 9 (a) shows that the average latency for verifying one vehicle in accumulator is about 0.157ms and decreases with number of authorized certificates increasing at an exponential rate, which is only slightly affected by the change of certificates. When verifying 1000 vehicles in the same time, the average latency is 9.1ms as shown in Figure 9 (b) with 10^6 authorized certificates in EPAM.

We can safely conclude that with the asynchronous accumulator assists, the membership verification overhead in the authentication phase decreases significantly.

6.4.3. *Communication overhead*

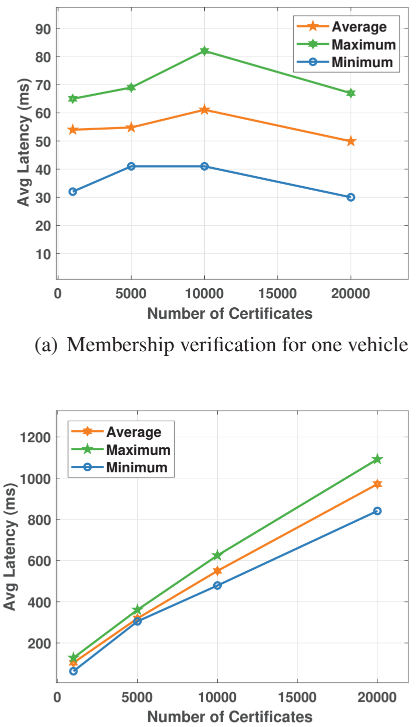
We analyze the communication overhead of EPAM, comparing with CPAS, ABAKA, and IBV.

We first calculate the message size based on the cryptography used in our scheme. The message size of Msg_1 is 85 bytes. Msg_2 sent by the vehicles to an RSU is 348 bytes, whereas Msg_3 sent from RSU to RSM is 369 bytes. Table 4 lists the comparison of message size, which shows that message size in EPAM is bigger than the other schemes. That's because Msg_2 are encrypted and randomized by adding timestamp as well as random numbers.

6.4.4. *Authentication latency*

Based on the message size, we conduct the simulation for authentication latency. We define the authentication latency as the moment the vehicle receives Msg_1 to when the vehicle is successfully authenticated. Figure 10 shows the relationship between the average latency and the number of vehicles. In general, the average message latency increases with the number of vehicles increasing. Our scheme performs better than CPAS, ABAKA, and IBV, even with bigger message size. Because the accumulator reduces the latency for membership verification comparing with the standard blockchain-based schemes. Comparing with IBV, whose latency increases from 25ms to 150ms when the number increases from 25 to 200, our scheme needs only 29.8ms.

We observe that the average latency of BPPA is 59 ms for authenticating 100 vehicles, and our scheme needs 19.7 ms. Therefore, EPAM achieves higher efficiency in the authentication.



(a) Membership verification for one vehicle

(b) Membership verification for 1000 vehicles

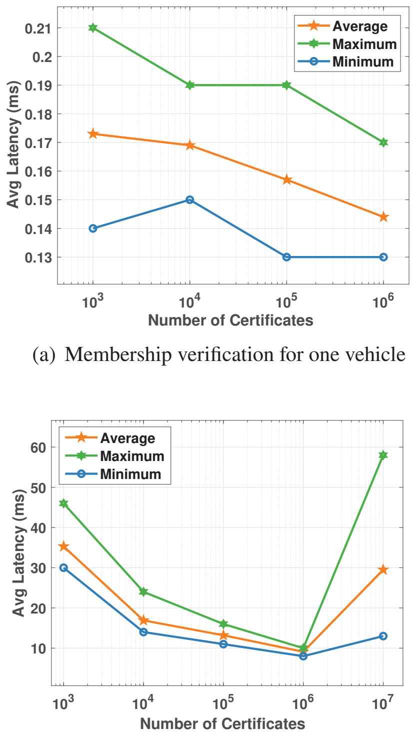
Figure 8: Performance results via Hyberledger Fabric simulations for membership verification in terms of the relationship between the latency and the number of vehicles.

7. Conclusion and future work

Our scheme leveraged blockchain technology and extended its structure with the asynchronous accumulators to achieve higher efficient and privacy-preserving authentication in VANETs. In addition, our scheme provided fault tolerance to resist semi-trusted RSUs. Moreover, we analyzed the computation overhead and compared it with the other authentication schemes. Simulations showed that our scheme is a promising efficient authentication scheme. The future work would focus on implementing the model in an automated way and allow it to interface with real-world data. This is a concrete step towards practical applications.

8. Acknowledgments

This document is results of the research project funded by the National Natural Science Foundation of China under NO. 61902157 and 62002139, China Postdoctoral Science Foundation under NO. 2019M661753 and 2019M651738, and the Natural Science Foundation of Jiangsu Province under NO. BK20200886. Xia Feng focused on conceptualization of this study and original draft preparation. Qichen Shi worked on simulation of Hyberleger Fabric, software and experiment. Qingqing Xie worked on the revision of the paper. Lu Liu is responsible for data curation and methodology.



(a) Membership verification for one vehicle

(b) Membership verification for 1000 vehicles

Figure 9: Performance results via asynchronous accumulator for membership verification in terms of the relationship between the latency and the number of certificates.

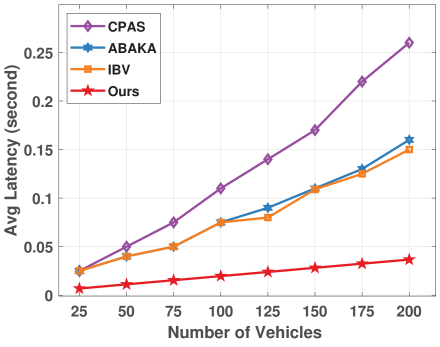


Figure 10: Performance comparison via simulations for different authentication schemes in terms of the relationship between the average latency and the number of vehicles.

References

[1] B. Cao, J. Zhao, P. Yang, Y. Gu, K. Muhammad, J. J. Rodrigues, and V. H. C. de Albuquerque, "Multiobjective 3-d topology optimization of next-generation wireless data center network," IEEE Transactions on Industrial Informatics, vol. 16, no. 5, pp. 3597–3605, 2019.

[2] D. Anadu, C. Mushagalusa, N. Alsbou, and A. S. Abuabed, "Internet of things: Vehicle collision detection and avoidance in a vanet environment," in IEEE International Instrumentation and Measurement Technology Conference. IEEE, 2018, pp. 1–6.

[3] M. Gerlach and F. Guttler, "Privacy in vanets using changing pseudonyms-ideal and real," in IEEE 65th Vehicular Technology Conference. IEEE, 2007, pp. 2521–2525.

[4] Y. Liu, L. Wang, and H.-H. Chen, "Message authentication using proxy

vehicles in vehicular ad hoc networks," IEEE Transactions on vehicular technology, vol. 64, no. 8, pp. 3697–3710, 2014.

[5] Y. Hao, Y. Cheng, and K. Ren, "Distributed key management with protection against rsu compromise in group signature based vanets," in IEEE GLOBECOM Telecommunications Conference, 2008, pp. 1–5.

[6] J. Zhang, S. Zhong, T. Wang, H.-C. Chao, and J. Wang, "Blockchain-based systems and applications: a survey," Journal of Internet Technology, vol. 21, no. 1, pp. 1–14, 2020.

[7] Y. Yao, X. Chang, J. Mišić, V. B. Mišić, and L. Li, "Bla: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3775–3784, 2019.

[8] 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 Systems, vol. 27, no. 12, pp. 2792–2801, 2019.

[9] A. Yang, J. Weng, K. Yang, C. Huang, and X. Shen, "Delegating authentication to edge: A decentralized authentication architecture for vehicular networks," IEEE Transactions on Intelligent Transportation Systems, 2020.

[10] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, and H. Wang, "Privacy-preserving cloud-based road condition monitoring with source authentication in vanets," IEEE Transactions on Information Forensics and Security, vol. 14, no. 7, pp. 1779–1790, 2018.

[11] L. Reyzin and S. Yakoubov, "Efficient asynchronous accumulators for distributed pki," in International Conference on Security and Cryptography for Networks. Springer, 2016, pp. 292–309.

[12] C. Fromknecht, D. Velicanu, and S. Yakoubov, "Certcoin: A namecoin based decentralized authentication system 6.857 class project," Unpublished class project, 2014.

[13] Namecoin, "Namecoin," https://namecoin.info., 2010.

[14] B. Laurie, "Certificate transparency," Communications of the ACM, vol. 57, no. 10, pp. 40–46, 2014.

[15] J. Liu, X. Li, Q. Jiang, M. S. Obaidat, and P. Vijayakumar, "Bua: A blockchain-based unlinkable authentication in vanets," in IEEE International Conference on Communications. IEEE, 2020, pp. 1–6.

[16] J. Benaloh and M. De Mare, "One-way accumulators: A decentralized alternative to digital signatures," in Advances in Cryptology — EUROCRYPT '93. Springer, 1993, pp. 274–285.

[17] T. Wang, Y. Mei, X. Liu, J. Wang, H.-N. Dai, and Z. Wang, "Edge-based auditing method for data security in resource-constrained internet of things," Journal of Systems Architecture, vol. 114, p. 101971, 2021.

[18] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in International Workshop on Peer-to-Peer Systems. Springer, 2002, pp. 53–65.

[19] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, "Amoeba: Robust location privacy scheme for vanet," IEEE Journal on Selected Areas in communications, vol. 25, no. 8, pp. 1569–1589, 2007.

[20] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," Journal of computer security, vol. 15, no. 1, pp. 39–68, 2007.

[21] M. D. Ryan, "Enhanced certificate transparency and end-to-end encrypted mail." in NDSS, 2014, pp. 1–14.

[22] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in Proceedings of the 13th EuroSys conference, 2018, pp. 1–15.

[23] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," SIGCOMM demonstration, vol. 14, no. 14, p. 527, 2008.

[24] I. D. Chakeres and E. M. Belding-Royer, "Aodv routing protocol implementation design," in 24th International Conference on Distributed Computing Systems Workshops. IEEE, 2004, pp. 698–703.

[25] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. IEEE, 2018, pp. 264–276.

[26] M. C. Library, "Multiprecision integer and rational arithmetic c/c++ library," URL: https://github.com/miracl/core, 2020.

[27] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in IEEE INFOCOM 2008-The 27th Conference on Computer Communications. IEEE, 2008, pp. 246–250.

[28] J. Huang, L. Yeh, and H. Chien, "Abaka: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks," IEEE Transactions on Vehicular Technology, vol. 60, no. 1, pp. 248–262, 2010.

[29] K.-A. Shim, "Cpas: an efficient conditional privacy-preserving authentication scheme for vehicular sensor networks," IEEE Transactions on Vehicular Technology, vol. 61, no. 4, pp. 1874–1883, 2012.

Xia Feng received the B.S. degree in computer science and technology from Jiangsu University in 2008, and the Ph.D. degree in computer science and technology department from Anhui University in 2017. She currently works as a associate professor with the School of automotive and traffic engineering, Jiangsu University, Zhenjiang, China. Her current research interests include authentication protocols in IoT blockchain and applied cryptography.

Qichen Shi is currently working toward the M.E. degree in computer science and communication engineering, Jiangsu University. His research interests is security in vehicular ad-hoc network.

Qingqing Xie received the master degree and the PhD degree in computer science and technology department from Anhui University, Hefei, China, in 2012 and 2017, respectively. She currently works as a lecturer with the school of computer science and communication engineering, Jiangsu University, Zhenjiang, China. Her research interests include blockchain, cloud computing and applied cryptography.

Lu Liu received the MSc degree in data communication systems from Brunel University, United Kingdom, and the PhD degree from the University of Surrey, United Kingdom. He is the professor of Informatics and head of Department of Informatics in the University of Leicester, United Kingdom. His research interests include areas of cloud computing, service computing, computer networks and peer-to-peer networking. He is a fellow of British Computer Society (BCS).