

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Citation:

Xu, Shiwei, Tao Wang, Ao Sun, Yan Tong, Zhengwei Ren, Rongbo Zhu, and Houbing Herbert Song. "Post-Quantum Anonymous, Traceable and Linkable Authentication Scheme Based on Blockchain for Intelligent Vehicular Transportation Systems." IEEE Transactions on Intelligent Transportation Systems, 2024, 1–12. <https://doi.org/10.1109/TITS.2024.3383668>.

DOI:

<https://doi.org/10.1109/TITS.2024.3383668>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

**Please provide feedback**

Please support the ScholarWorks@UMBC repository by emailing [scholarworks-group@umbc.edu](mailto:scholarworks-group@umbc.edu) and telling us what having access to this work means to you and why it's important to you. Thank you.

# Post-Quantum Anonymous, Traceable and Linkable Authentication Scheme Based on Blockchain for Intelligent Vehicular Transportation Systems

Shiwei Xu<sup>1</sup>, Tao Wang<sup>1</sup>, Ao Sun, Yan Tong<sup>2</sup>, *Member, IEEE*, Zhengwei Ren, Rongbo Zhu<sup>3</sup>, *Member, IEEE*, and Houbing Herbert Song<sup>4</sup>, *Fellow, IEEE*

**Abstract**—As the Internet of Vehicles (IoV) has become the critical part of Intelligent Vehicular Transportation Systems (IVTS), massive IoV entities (e.g., RSU, OBU, pedestrians' mobile devices, etc.) get involved into IVTS. At present, one of the biggest challenges with IoV/IVTS is how to maintain a balance between security and privacy. The receivers need to be sure that they are receiving reliable messages from the origin and could trace or link the attacker's identity, but the tracing or linking may work against the sender's need for identity privacy. To solve the security and privacy problem, most of current works have proposed authentication solutions to provide anonymous, traceable and unlinkable schemes, which are still vulnerable to either Sybil attacks or quantum attacks. Therefore, we propose the blockchain-based post-quantum anonymous, traceable and linkable authentication scheme by utilizing NIST winner post-quantum algorithms and related post-quantum linkable ring signature. Grounded on the authentication scheme, we also develop key exchange mechanism, which help IoV entities perform efficient message authentication encryption/decryption during P2P communication and broadcast. The security analysis shows that our proposal is resistant to Sybil attack and provides other essential security characteristics including man-in-the-middle-proof and anti-replay. Finally, we perform detailed performance evaluation including each on-chain API execution time, the off-chain communication time and the on-board/on-chain storage requirements. To further evaluate the feasibility of our scheme in the IoV/IVTS environment, we also show the effectiveness of our proposal in a blockchain-based simulation study.

**Index Terms**—Post-quantum, security and privacy, IoV/IVTS authentication, key exchange, blockchain-based.

## I. INTRODUCTION

INTELLIGENT Vehicular Transportation Systems (IVTS) enables seamless communication between vehicles, pedestrians, road side units, etc., and Internet of Vehicles (IoV), which currently is the critical part of IVTS, allows vehicles to communicate in real time with other vehicles, roadside infrastructure, and pedestrians, known as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I) and Vehicle-to-Pedestrian (V2P). Moreover, IoV also helps one vehicle to understand its internal and roadside conditions by monitoring its onboard sensors and recognizing the road signs, which are known as Vehicle-to-Sensors (V2S) and Vehicle-to-Roadside (V2R). Based on the architecture, the IoV/IVTS communication system consists of several types of network nodes which mainly include the onboard unit (OBU) in the vehicle, the roadside unit (RSU) in the roadside infrastructure, pedestrians' mobile devices and so on. With the rapid increase in the number of vehicles and pedestrians' mobile devices, IoV has become one of the hottest research topics in Intelligent Vehicular Transportation Systems [1], [2], [3].

Authentication mechanism can help network nodes in IoV/IVTS (e.g., RSU, OBU, pedestrians' mobile devices and so on) to identify legitimate/malicious nodes, and is considered to be the first line of defense against various attacks in IoV/IVTS [4]. However, the network layer of the vehicular communications provide no authentication and poor privacy mechanism, in which only the IP address of each vehicle is dynamically changed [5].

The most common way to guarantee the authenticity of vehicles and messages in IoV/IVTS is to adopt certificates and Public Key Infrastructure (PKI)/Certification Authority (CA), and in order to preserve the identity privacy of vehicles in IoV/IVTS, CA would allow one vehicle to generate multiple certified identities (i.e., public/private key pairs). In each public key certificate from the same vehicle, one short-term pseudonym (instead of real identity information) is given by the CA for the vehicle to hide its real identity. Since one vehicle can have multiple certified identities, the pseudonym mechanism is vulnerable to Sybil attack [6], during which one

Manuscript received 24 July 2023; revised 31 January 2024; accepted 18 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61902285 and Grant 62004077; in part by the Knowledge Innovation Program of Wuhan-Basic Research under Grant 2022010801010225; in part by the Open Foundation of Hubei Key Laboratory of Applied Mathematics, Hubei University, under Grant HBAM202101; and in part by the Fundamental Research Funds for the Central Universities under Grant 2662022XXYJ004. The Associate Editor for this article was Z. Liu. (*Corresponding author: Rongbo Zhu.*)

Shiwei Xu and Rongbo Zhu are with the Engineering Research Center of Intelligent Technology for Agriculture, Ministry of Education, and the College of Informatics, Huazhong Agricultural University, Wuhan 430070, China (e-mail: rbzhu@mail.hzau.edu.cn).

Tao Wang and Yan Tong are with the College of Informatics, Huazhong Agricultural University, Wuhan 430070, China.

Ao Sun is with Wuhan Maritime Communication Research Institute, Wuhan 430079, China.

Zhengwei Ren is with the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430081, China.

Houbing Herbert Song is with the Department of Information Systems, University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250 USA. Digital Object Identifier 10.1109/TITS.2024.3383668

vehicle can use multiple identities simultaneously to pretend to be multiple vehicles and launch more serious attacks like Denial of Service (DoS) attack.

To reduce or eliminate the burdensome certificate management work in the PKI-based scheme, identity-based signature (IBS) [7] and certificateless signature (CLS) [8] have been proposed. In IBS-based and CLS-based authentication schemes [9], [10], [11], [12], the IoV/IVTS node's identifier is used as the public key (or to generate the public key) and the messages are signed by the private key (or together with some identifier-related secret value) generated from the identifier. However, since the identifier of the corresponding IoV/IVTS node is needed during the signature verification of IBS and CLS, pseudonym/randomness mechanism is still widely used to protect the identity privacy of IoV/IVTS nodes and thus makes the IBS/CLS scheme vulnerable to Sybil attack and the followed attacks.

Another solution to preserve the identity privacy of IoV/IVTS node is using group/ring-based signature scheme [13], where messages are signed by valid group/ring members on behalf of the whole group/ring and only the group/ring manager has the capability to identify who is the actual sender [14], [15], [16], [17], [18]. Nevertheless, the group/ring-based signature scheme also cannot resist the Sybil attack because normal nodes in IoV/IVTS cannot distinguish two IoV/IVTS nodes in the same group/ring. To keep balance between the privacy and security in IoV/IVTS authentication, only few work [19] proposes IoV/IVTS authentication scheme based on linkable group/ring signature, which allows anyone in the IoV/IVTS to check authentication and linkability between multiple pseudonyms of one IoV/IVTS node without revealing the node's real identity.

Although the linkable group/ring signature-based scheme keeps a good balance between privacy and security (i.e., resistant to Sybil attack), current related scheme heavily relies on traditional asymmetric cryptography that is based on mathematics problems of integer factorization, discrete logarithm or bilinear pairing. If a practical quantum computer is available, then the Shor's algorithm would solve the related mathematics problems in polynomial time. In that case, the security assumption of linkable group/ring signature based on traditional asymmetric cryptography would be invalid, and this makes the related authentication schemes insecure against the quantum computer.

To address the quantum-safe issue, the National Institute of Standards and Technology (NIST) initiates the call for standard proposals of post-quantum public key cryptosystems including digital signature algorithms and Key Encapsulation Mechanism (KEM) algorithms, which quite recently has announced its three digital signature and one KEM winner algorithms [20]. For quantum-safe consideration, it is meaningful to design, implement and evaluate the post-quantum linkable group/ring signature-based authentication scheme for IoV/IVTS based on the recognized winner algorithms. Furthermore, with the support of the quantum-safe identity authentication scheme for IoV/IVTS, the post-quantum key exchange mechanism can naturally be designed based on the KEM winner algorithm, which can be used to further support

the data authentication and data encryption/decryption during the IoV/IVTS communication.

Last but not least, the biggest disadvantage of post-quantum signature and KEM algorithms for IoV/IVTS is that most of the algorithms have very large size of public keys, which would bring substantial management work of the related certificates/keys and huge communication overhead between IoV/IVTS nodes. To overcome such a barrier, permissioned blockchain could be introduced to provide integrity and traceability services, which help simplify the certificate/key management and further reduce communication overhead [21].

### A. Our Contributions

Therefore, we propose the first security-privacy-balanced authentication scheme for IoV/IVTS based on NIST post-quantum winner algorithms, related post-quantum linkable ring signature and permissioned blockchain, which provides anonymous, traceable and linkable identity and data authentication services for vehicles. Benefited from the provided post-quantum authentication services, the scheme can guarantee the real-identity privacy of the nodes in IoV/IVTS, and at the same time is resistant to the Sybil attack and the followed attacks.

Our main contributions can be summarized as follows.

- Based on NIST post-quantum winner signature algorithms and related post-quantum linkable ring signature, we propose blockchain-based post-quantum authentication scheme that preserves the identity privacy of IoV/IVTS's nodes and at the same time provides identity linkability in order to protect against Sybil attacks that can be used to launch powerful attacks like DoS that threatens the service availability. We utilize the blockchain to make the key/certificate management easier.
- Grounded on the privacy-preserving and linkable authentication scheme, we also design and develop the corresponding post-quantum blockchain-based key exchange mechanism for IoV/IVTS nodes, which is also based on NIST post-quantum winner KEM algorithms. With the successful on-chain key exchange, IoV/IVTS nodes can perform efficient message authentication encryption/decryption during P2P communication and broadcast.
- We perform detailed performance evaluation of each on-chain API execution time, the off-chain communication time and the storage requirements. To further evaluate the feasibility of our authentication scheme in the IoV/IVTS environment, we show the effectiveness of our proposal in a blockchain-based simulation study.

### B. Paper Organization

The rest of the paper is structured as follows. The related works of this paper are introduced, compared and summarized in Section II. In Section III, we overview all the associated preliminaries. And then, we present how we design and implement our IoV/IVTS authentication scheme in Section IV. The security analysis is given in Section V, and in Section VI we perform a detailed performance evaluation and

blockchain-based simulation of our proposed scheme. Finally, the conclusion is drawn in the Section VII.

## II. RELATED WORKS

### A. Blockchain-Based Authentication Mechanisms for IoV/IVTS

In recent years, blockchain has been extensively used to improve the security, privacy and trust management in vehicular networks [22], [23]. The work in [24] provides a blockchain-enabled certificate-based IoV/IVTS authentication mechanism that improves the transparency of certificate management and eliminates certificates revocation list. Moreover, many certificate-based authentication mechanisms [25], [26], [27] have been designed to preserve the vehicles' identities conditional privacy and unlinkability.

To reduce the certificates management work, authors propose multiple blockchain-enabled ID-based [9], certificateless-signature-based [10], [11] and reputation-based [28], [29] authentication schemes for vehicular networks. However, since the identifier of the corresponding IoV/IVTS node is needed during the signature and reputation verification, pseudonym/randomness mechanism is still used to protect the identity privacy of IoV/IVTS nodes and thus makes all the above schemes vulnerable to Sybil attack and followed attacks.

A few researchers [14], [15] make use of group/ring signature to develop the blockchain-based authentication schemes for vehicular networks, where pseudonyms are given to the IoV/IVTS nodes in the group/ring to provide and only the group/ring manager is aware of the linkability between the real identities and the pseudonyms. Therefore, as we can see, all the above blockchain-based authentication mechanisms for vehicular networks cannot keep a good balance between identity privacy and security (i.e., resistant to Sybil attack) and are based on traditional cryptographic assumptions which make them all vulnerable to attacks from quantum computers.

### B. Post-Quantum Authentication Mechanisms for IoV/IVTS

Quantum-safe is another hot security issue in IoV/IVTS authentication for the past few years [30]. Some early work [31] proposes post-quantum PKI-based authentication scheme grounded on lattice-based cryptography and only provides basic post-quantum CA-services like certificate generation, pseudonym generation, ID-pseudonym linkage and misbehavior tracing. After that, various works have been proposed to protect the identity privacy in vehicular networks by making use of post-quantum identity-based authentication [12], trust-based authentication [32] and group/ring signature [16], [17], [18]. However, all of these above works do not utilize blockchain to simplify the certificates/keys management, and do not deal with the problem of identity linkability.

### C. Blockchain-Based Post-Quantum Authentication Mechanisms for IoV/IVTS

Only few work makes use of blockchains to develop post-quantum authentication mechanisms for IoV/IVTS. The authors of [33] develop blockchain-envisioned

multivariate-based multi-signature scheme, in which one aggregated signature signed by multiple signers can be verified by one verifier. But the work [33] does not deal with many important security issues such as identity anonymity, traceability and (un)linkability in IoV/IVTS.

The closest work to ours is [34], where the authors make use of lattice cryptography and blockchain to provide IoV data authentication mechanism supporting plenty of security characteristics such as identity/message authentication, conditional anonymity, traceability, unlinkability, resistance of quantum attack and so on. However, since the work [34] uses randomness to hide linkage between two signatures from the same vehicles, no entity except the (semi-)trusted third party can link two randomized signatures from one malicious vehicle and thus the scheme is also vulnerable to Sybil attack and the followed attacks. Finally, we sum up and compare the related researches with our work in Table I.

## III. SYSTEM BUILDING BLOCKS

### A. PQ Algorithms in the NIST Call

At the end of the third round of the NIST call for PQ algorithms, there are three digital signature (i.e., CRYSTALS-DILITHIUM, FALCON, SPHINCS+) and one KEM (i.e., CRYSTALS-KYBER) winner algorithm, which have been selected as the national standards of US. Moreover, three candidate KEM algorithms (i.e., HQC, BIKE, Classic McEliece) are potentially standardized in the ongoing fourth round. By using different key sizes and parameter sets, all the algorithms can achieve different NIST security levels (i.e., bits-of-security level), which is defined as the effort required by a classical computer to perform a brute-force attack on a given-length cryptographic key. Normally NIST security levels 1~5 approximately imply 128/160/192/224/256-bits-of security levels. One key encapsulation mechanism *KEM* consists of following 3 steps.

- The key generation step  $\text{KEM.KeyGen}()$ , that outputs a public/private key pair  $(pk_{kem}, sk_{kem})$ .
- The encapsulation step  $\text{KEM.Encap}(pk_{kem})$  that takes a public key  $pk_{kem}$  as input, and outputs a shared secret/ciphertext pair  $(SS, CT)$ .
- The decapsulation step  $\text{KEM.Decap}(sk_{kem}, CT)$ , that takes the corresponding private key  $sk_{kem}$  and the ciphertext  $CT$  as input, and outputs the shared secret  $SS$ .

### B. Kyber-Based Linkable Ring Signature

Based on the only winner KEM algorithm (i.e., CRYSTALS-KYBER), the Linkable Ring Signature Scheme with Stealth Addresses (SALRS) [35] has been recently proposed and consists of the following functions.

- $\text{MasterKeyGen}(seed) \rightarrow (MPK, MSK)$ . This function needs one randomly-generated number  $seed$  as the input, and outputs a master public/private key pair  $(MPK, MSK)$ . If  $seed_1 = seed_2$ , then the outputs of the  $\text{MasterKeyGen}(seed_1)$  and  $\text{MasterKeyGen}(seed_2)$  will be the same.
- $\text{DerivedPublicKeyGen}(MPK) \rightarrow DPK$ . It is a probabilistic function, which needs input a master public key  $MPK$  and would output a derived public key  $DPK$ .



TABLE I  
SECURITY CHARACTERISTICS COMPARISON OF MAIN RELATED WORKS

Related works	ID anonymity	ID traceability	ID linkability	Msg. authentication	Post-quantum	Blockchain-based
Wu et al. [19]	✓	✓	✓	✓	✗	✗
[14], [15]	✓	✓	✗	✓	✗	✓
[16]–[18]	✓	✓	✗	✓	✓	✗
Gupta et al. [33]	✗	✗	✗	✓	✓	✓
Gupta et al. [34]	✓	✓	✗	✓	✓	✓
Our work	✓	✓	✓	✓	✓	✓

- $\text{DerivedPublicKeyOwnerCheck}(DPK, MPK, MSK) \rightarrow 1/0$ . On input a derived public key  $DPK$  and a master public/private key pair  $(MPK, MSK)$ , the function outputs a bit  $b \in \{0,1\}$ , with  $b = 1$  meaning that  $DPK$  is a valid derived public key generated from  $MPK$  and  $b = 0$  otherwise.
- $\text{Sign\_LRS}(M, R, DPK, (MPK, MSK)) \rightarrow \sigma$ . This function needs the input of a message  $M$ , a ring of derived public keys  $R = (DPK_1, \dots, DPK_r)$ , a derived public key  $DPK \in R$ , and the master key pair  $(MPK, MSK)$  for  $DPK$ , and then the function outputs a signature  $\sigma$  on the message  $M$  based on the ring  $R$ .
- $\text{Verify\_LRS}(M, R, \sigma) \rightarrow 1/0$ . The function needs to be fed with a message  $M$ , a ring of public keys  $R$ , and a purported signature  $\sigma$  on the message  $M$  based on the ring  $R$ , and then the function will output a bit  $b \in \{0,1\}$ , with  $b = 1$  meaning that the signature  $\sigma$  is valid and  $b = 0$  means not.
- $\text{Link}(M_0, R_0, \sigma_0, M_1, R_1, \sigma_1) \rightarrow 1/0$ . This is a deterministic function, which needs the inputs of two valid signatures  $(M_0, R_0, \sigma_0)$ ,  $(M_1, R_1, \sigma_1)$ , and then the function will output a bit  $b \in \{0,1\}$ , with  $b = 1$  meaning the two signatures are linked and  $b = 0$  means not.

As a linkable ring signature, the scheme must satisfy the following correctness property.

- For any  $\text{MasterKeyGen}() \rightarrow (MPK, MSK)$ ,  $\text{DerivedPublicKeyGen}(MPK) \rightarrow DPK$ , it holds that  $\text{DerivedPublicKeyOwnerCheck}(DPK, MPK, MSK) = 1$ .
- For any message  $M \in \mathbf{M}$ , any ring of public keys  $R$ , and any  $DPK_r \in R$  such that  $\text{DerivedPublicKeyOwnerCheck}(DPK_r, MPK, MSK) = 1$  for some master key  $(MPK, MSK)$ , it holds that  $\text{Verify\_LRS}(M, R, \text{Sign\_LRS}(M, R, DPK_r, MPK, MSK)) = 1$ .
- For any messages  $M_0, M_1 \in \mathbf{M}$ , any rings  $R_0, R_1$ , and any  $DPK_{r_0} \in R_0, DPK_{r_1} \in R_1$  such that  $\text{DerivedPublicKeyOwnerCheck}(DPK_{r_i}, MPK_i, MSK_i) = 1$  for some master key  $(MPK_i, MSK_i)$  ( $i = 0, 1$ ), let  $\text{Sign\_LRS}(M_i, R_i, DPK_{r_i}, MPK_i, MSK_i)$  ( $i = 0, 1$ )  $\rightarrow \sigma_i$ . It holds that  $\text{Link}(M_0, R_0, \sigma_0, M_1, R_1, \sigma_1) = 1$  if  $DPK_{r_0} = DPK_{r_1}$ , and  $\Pr[\text{Link}(M_0, R_0, \sigma_0, M_1, R_1, \sigma_1) = 0] \geq 1 - \text{negl}(\lambda)$  if  $DPK_{r_0} \neq DPK_{r_1}$ , where  $\text{negl}$  is a negligible function.

### C. Consortium Blockchain and Chaincodes

Among current existing consortium blockchains, Hyperledger Fabric [36] is one of the most successful projects, and all the chaincodes should be installed on specific peers before the execution of Hyperledger Fabric. Only the organization administrators are able to modify the chaincodes installed

on the peers. The peers can invoke the chaincodes in order to upload data to on-chain ledgers (e.g., provide vehicle's information), perform online calculation (e.g., generate keys and certificates for the vehicle) and download on-chain data (e.g., download vehicle certificate). The APIs natively provided by Hyperledger Fabric to upload/download data are  $\text{PutState}()/\text{GetState}()$  in the package `shim`. While the network nodes outside the consortium need to log in as clients and can only invoke the APIs provided by chaincodes via client codes.

## IV. SCHEME DESIGN

### A. IoV/IVTS Authentication Security Requirements

In order to keep a balance between security and privacy, the following security requirements must be satisfied during authentication in IoV/IVTS.

- Identity anonymity. The vehicle's real identity (e.g., license plate number) can only be managed by authorized party like Registration Authority (RA), and no other entities in IoV know the real identity of the vehicle.
- Identity traceability: The RA can trace the real identity of any vehicle in the case that any rogue entity appears during the IoV communication. For example, the RA can identify a malicious node launching Denial of Service (DoS) attack by examining the transmitted messages.
- Identity (un)linkability: For privacy consideration, no malicious entity can link a vehicle's real identity to the vehicle's pseudonyms (or randomized identities). While for security consideration, one vehicle's pseudonyms (or randomized identities) can be linked to prevent Sybil attack and the followed attacks (e.g., DoS attack).
- Key exchange and message authentication: With the support of identity authentication, there should be an efficient way for IoV entities to exchange key, which could be further used to authenticate the received messages.
- Post-quantum authentication: The identity/message authentication mechanisms in IoV communication should be resistant to quantum attacks, i.e., the protocol must resist a cryptanalytic attack by quantum computers.
- During the identity/message authentication, the protocol should also resist other threats such as man-in-the-middle, message modification, replay attacks and so on.

### B. System Architecture and Topology Based on Blockchain

Besides the above security requirements of authentication in IoV/IVTS, we design our post-quantum certificate-based scheme grounded on consortium blockchain (i.e., Hyperledger Fabric), which is much faster in transaction processing than the public blockchains (e.g., Ethereum). In Fabric, all the

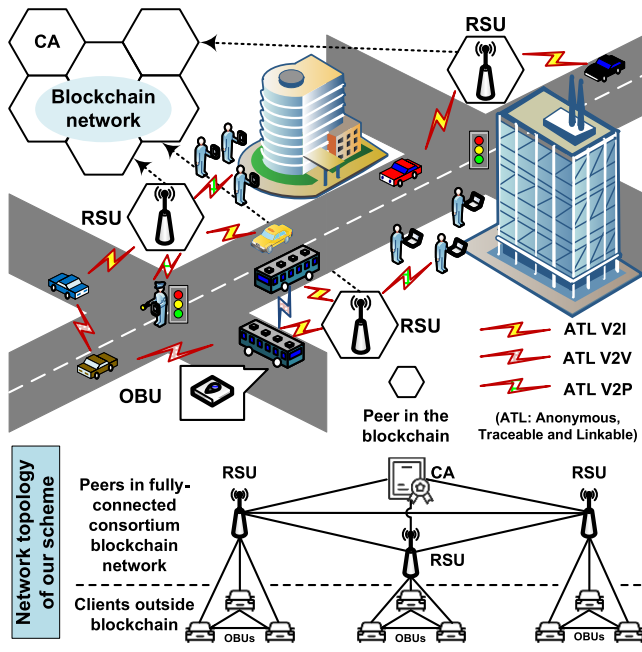


Fig. 1. The architecture and network topology of our blockchain-based anonymous, traceable and linkable authentication scheme for IoV/IVTS.

peers (e.g., CA and RSU nodes) can also be configured fully connected and the routings between peers are automatically managed by Fabric, which makes the CA/RSU network more autonomous. Moreover, each peer in Fabric maintains a copy of append-only public certificate-related information, and thus the whole certificate system is more robust and immutable.

The system architecture and network topology of our authentication scheme is shown in Figure 1, where multiple RSUs and one CA are configured as fully-connected blockchain peers, which are responsible for certificate management. While, all the other IoV entities (e.g., the OBUs in the vehicles and pedestrians' mobile devices) are considered as the client entities outside the blockchain network, and are excluded from certificate management.

### C. The Proposed Authentication Scheme

The execution flow of our proposed authentication scheme consists of seven main steps: 1) *system initialization*, 2) *keys/certificates generation*, 3) *IoV client entities join RSU*, 4) *P2P communication in V2V/V2P*, 5) *Broadcast in V2V/V2P*, 6) *keys/certificates update/revocation* and 7) *IoV client entities leave RSU*.

1) *System Initialization*: We deploy all the peers of the blockchain network, where the CA and multiple RSUs are configured as the peers in the blockchain network, while all the OBUs and pedestrians' mobile devices are considered as the client nodes of the blockchain network. Then, we install three chaincodes (i.e., RSU chaincode, Client chaincode and CA chaincode) on the corresponding peers, among which the RSU peer is installed with both RSU and Client chaincodes because OBUs and other mobile devices need to connect to one RSU peer as clients and then invoke the Client chaincode to interact with other chaincodes.

2) *Keys/Certificates Generation and Offline Registration*: This is a setup phase, which is executed by *IoV entities*

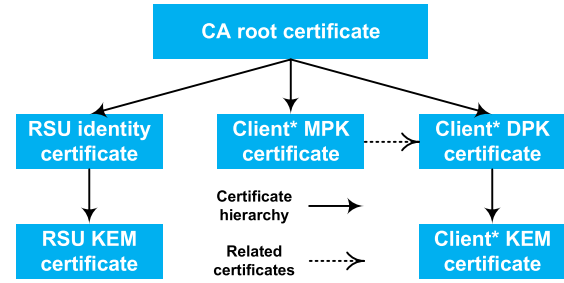


Fig. 2. The certificates hierarchy of our blockchain-based ATL authentication scheme for IoV. The client\* in the figure is short for all the IoV client entities (i.e., OBUs, pedestrians' mobile devices and so on) in our scheme.

(e.g., RSU, OBU, pedestrians' mobile devices, etc.) so as to proactively register in the Certificate Authority (CA) to join the IoV network. As the CA and RSUs, there is no need to hide their identities, so the NIST post-quantum digital signatures are used to generate identity key pairs for them. During the deployment of the RSU, one RSU peer randomly generates one public/private key pair  $(pk_{id}, sk_{id})$  as identity key pair and sends the public key  $pk_{id}$  to CA peer in order to register as a legal RSU in IoV.

While in order to provide (un)linkable identity service for *IoV client entities* (i.e., OBU and pedestrians' mobile devices), the post-quantum linkable ring signature algorithms are used to generate the identity key pairs for IoV client entities. Each of the IoV client entities should randomly choose a private seed  $seed_A$ , which is further used to generate its master public/private key pair  $(mpk_A, msk_A)$ , i.e.,  $MasterKeyGen(seed_A) \rightarrow (mpk_A, msk_A)$ . Then, the IoV client entity sends its master public key  $mpk_A$  together with its detailed information (about the OBU, vehicle or pedestrian) to CA peer in order to register as a legal client entity in IoV. After offline verification of the detailed information, the CA peer would help the IoV client entity generate its derived public key  $dpk_A$  using the master public keys  $mpk_A$  provided by the IoV client entity, i.e.,  $DerivedPublicKeyGen(mpk_A) \rightarrow dpk_A$ .

Based on all the received/generated public keys, the CA peer would generate and issue a set of identity certificates for the IoV entities, which are shown in Figure 2. The top of the certificates hierarchy is the CA root certificate, whose child certificates include RSU identity certificate (i.e.,  $Cert_{rsu}$ ), the IoV client entity's identity certificate for its  $mpk_A$  and anonymous certificate for its  $dpk_A$  (i.e.,  $Cert_{mpk_A}$  and  $Cert_{dpk_A}$ ). At the same time, it is allowed to have only one valid identity certificate for each RSU, IoV client entity's  $mpk_A$  and  $dpk_A$ . Based on the identity certificates, the RSUs and IoV client entities could request certificates for their KEM public keys (e.g.,  $Cert_{kem}$ ), which are used to perform key exchange with each other in order to generate one session key  $sessKey$  protecting the transmitted data between IoV entities.

Thanks to the help of the blockchain system (i.e., Hyperledger Fabric), the CA peer can conveniently manage the on-chain certificates, which are stored under the certificate serial number  $serialNum$  generated and assigned by the CA and can be further queried/updated/revoked with a given certificate serial number.

Upon receiving the  $dpk_A$  and corresponding certificates from CA, the IoV client entity should perform

$\text{DerivedPublicKeyOwnerCheck}(dpk_A, mpk_A, msk_A)$  in order to make sure that  $dpk_A$  is a valid derived public key generated from its master public key  $mpk_A$ .

3) *IoV Client Entities Join RSU*: Once all the IoV entities are registered, the IoV system together with our proposed authentication mechanism is put into operation. Each RSU peer would maintain a ring of derived public keys (e.g.,  $dpk_1, dpk_2, \dots, dpk_i$ ) from all the IoV client entities connecting to the RSU, and initially the ring is empty.

When one newcomer IoV client entity gets into the communication range of one RSU, the client entity would firstly request to join the ring maintained by the RSU. And the join-ring request includes a join-ring message, a self-signed signature (by the newcomer IoV client) and the IoV client entity's  $mpk_A$  and  $dpk_A$  certificate serial numbers, which can be used to easily retrieve the corresponding certificate onchain.

After the receipt of the join-ring request, the RSU peer retrieves the corresponding  $mpk_A$  and  $dpk_A$  certificates of the IoV client entity, verifies the identity of the IoV client entity based on the retrieved certificates, and checks whether the join-ring request message is self-signed by the  $msk_A$ . If all the checks pass, then the RSU accepts the newcomer IoV client entity and adds its  $dpk_A$  into the ring of RSU. The main idea of join-ring chaincode design is briefly presented in the Algorithm 1.

Alg. 1: IoV client entity requests to join RSU ring

**Prerequisites:**

**IoV client entity:**  $mpk_A, msk_A, dpk_A, Cert_{mpk_A}, Cert_{dpk_A}$   
**RSU:**  $rsuRing$ ;

**function Join\_Ring()** in Client chaincode

```

  CertmpkA.serialNo → sNo1;
  CertdpkA.serialNo → sNo2;
  "Request to join RSU ring" → jrReqA;
  // The client entity makes a ring only based on its dpkA and signs
  the join-ring request
  {dpkA} → RingA;
  Sign_LRS(jrReqA, RingA, dpkA, (mpkA, mskA)) → σA;
  // Invoke the function Join_Ring() in RSU chaincode
  shim.InvokeChaincode("RSU", jrReqA, σA, sNo1, sNo2);

```

**function Join\_Ring()** in RSU chaincode

```

  // Upon receiving jrReqA, σA, sNo1, sNo2 from Client
  chaincode
  RetrieveCertFromCA(sNo1) → CertmpkA;
  RetrieveCertFromCA(sNo2) → CertdpkA;
  // The identity could be verified based on CertmpkA
  CertdpkA.dpk → dpkA;
  {dpkA} → RingA;
  if(Verify_LRS(jrReqA, RingA, σA) == 1)
  then AppendRing(rsuRing, dpkA);

```

4) *P2P Communication in V2V/V2P*: After joining the ring maintained by the RSU, one IoV client entity can communicate with other IoV entities (client entities or the RSU) in the same ring. In the case that two client entities (e.g.,  $A$  and  $B$ ) perform P2P communication with each other,  $A$  needs to firstly verify  $B$ 's DPK certificate  $Cert_{dpk_B}$  and the KEM certificate  $Cert_{kem_B}$  as shown in Figure 2 in order to make sure that  $B$  is one registered client entity with registered KEM public key  $pk_{kem_B}$ . And then,  $A$  performs key exchange with  $B$  based on the KEM mechanism  $\text{KEM.Encap}(pk_{kem_B}) \rightarrow$

$(sessKey_{AB}, cipherText_{AB})$ , among which  $cipherText_{AB}$  would be sent to  $B$ . After receiving  $cipherText_{AB}$ , DPK/KEM certificates from  $A$ , the client entity  $B$  would firstly do the same certificates verification as  $A$  does. If the verification succeeds, then  $B$  decapsulates the  $cipherText_{AB}$  by using its own KEM private key  $sk_{kem_B}$  and gets the session key  $sessKey_{AB}$ . Benefited from the on-chain Certificate Authority (CA), clients  $A$  and  $B$  can invoke the verification chaincode API by providing only a set of certificate serial numbers, and all the related DPK/KEM certificates could be verified. After successful key exchange,  $A$  and  $B$  could perform secure and efficient off-chain communication by using the session key  $sessKey_{AB}$  as the symmetric encryption/decryption (e.g., AES) key or the Message Authentication Code (MAC) key to protect the confidentiality, integrity and authentication of the transmitted message between  $A$  and  $B$ .

If the client entity  $A$  wants to perform secure P2P communication with the RSU, then both  $A$  and the RSU should perform the similar certificates verification and the same key exchange as mentioned in the above paragraph. The only difference during certificate verification is that the RSU is not anonymous and therefore only has identity certificate, which could be verified in the RSU certificate chain branch as shown in Figure 2. With a successful key exchange, the client entity  $A$  and the RSU could perform encrypted and authenticated off-chain P2P communication just like two IoV client entities.

5) *Broadcast in V2V/V2P*: The main idea of the IoV client entity broadcasting is briefly summarized in the Algorithm 2.

Alg. 2: IoV Client entity broadcasts in V2V/V2P

**Prerequisites:**

**Broadcasting client entity:**  $brdcstMsg, mpk_A, msk_A, dpk_A$   
**RSU:**  $rsuRing$ ;

**function Brdcst\_Init()** in Client chaincode

```

  // The client entity retrieves the latest ring from RSU
  shim.InvokeChaincode("RSU", "Retrieve ring") → rsuRing;
  // The client entity updates the retrieved ring on-chain using the
  ring hash value as the index
  Hash(rsuRing) → hRsuRing;
  shim.PutState(hRsuRing, {rsuRing});
  return rsuRing;

```

**function Broadcast()** in client codes

```

  // The client entity execute Brdcst_Init() in Client chaincode
  client.Execute("Client", "Brdcst_Init") → rsuRing;
  Sign_LRS(brdcstMsg, rsuRing, dpkA, (mpkA, mskA)) →
  σA;
  Hash(rsuRing) → hRsuRing;
  // Broadcast the message, signature and the RSU ring hash value
  socket.BrdcstOut(brdcstMsg, σA, hRsuRing);

```

**function Rcv\_Brdcst\_Init()** in Client chaincode

```

  // Retrieve the RSU ring based on given ring hash value
  shim.GetState(hRsuRing) → rsuRing;
  return rsuRing;

```

**function Rcv\_Broadcast()** in client codes()

```

  // Receive the broadcast message, signature and RSU ring hash
  value
  socket.BrdcstIn() → (brdcstMsg, σA, hRsuRing);
  // Retrieve the RSU ring from on-chain public ledger
  client.Execute("Client", "Rcv_Brdcst_Init", hRsuRing) →
  rsuRing;
  if(Verify_LRS(brdcstMsg, rsuRing, σA) == 1)
  then AcceptBrdcst(brdcstMsg);

```



When one IoV client entity needs to broadcast its message (e.g., its location, speed and so on) to other IoV entities, the broadcasting client entity should use its DPK to sign the broadcast message based on the latest ring maintained by the RSU. And the broadcast-receiving entity should verify the signature in order to make sure that the broadcasting entity is a legitimate IoV client entity accepted by the RSU and its DPK has also been added to the ring maintained by the RSU.

When one RSU wants to broadcast message to the connected client entities, the RSU only needs to sign the broadcast message by using its identity private key  $sk_{id}$ , and other client entities could verify the signature based on the RSU identity public key  $pk_{id}$  contained in the RSU certificate  $Cert_{rsu}$ .

6) *IoV Client Entities Update/Revoke Certificates*: If the private keys of one IoV client entity are leaked, the IoV client entity needs to send an off-chain update/revoke-certificate request to CA peer. Since all the certificates are managed by the CA peer onchain, the CA chaincode only needs to invalidate the old certificate and then generate a new one for the requesting IoV client entity (if update-certificate request received). Periodic updating the DPK certificate would also help to preserve the privacy of one IoV client entity.

When receiving an updated DPK certificate, one IoV client entity should update its DPK in the ring maintained by the RSU. The update-ring request from the IoV client entity (e.g., A) includes a self-signed message based on its revoked and updated DPK (i.e.,  $dpk_{A_{rvk}}$  and  $dpk_{A_{updt}}$ ) and the two DPK certificate serial numbers for the convenience of the RSU verifying the IoV client entity's original and updated identities. After obtaining the update-ring request, the RSU peer retrieves the corresponding two DPK certificates based upon the given serial numbers, gets the two DPKs  $dpk_{A_{rvk}}$  and  $dpk_{A_{updt}}$  from corresponding certificates, and checks the update-ring request is self-signed by A. Grounded on successful verification of the self signature, the RSU would remove  $dpk_{A_{rvk}}$  from its latest ring and add the updated DPK  $dpk_{A_{updt}}$  into its latest ring. The update-DPK chaincode design details are summarized in the Algorithm 3.

7) *IoV Client Entities Leave RSU*: At last, when one IoV client entity A leaves the communication range of one RSU, A should send the RSU a leave-ring request, which includes a leave-ring message  $msg_{lvRng}$ , its DPK  $dpk_A$  and a self-signed signature  $\sigma_A$  (based on  $dpk_A$  and  $Ring_A \leftarrow \{dpk_A\}$ ). The leave-ring request is stored onchain by A and can be retrieved by the RSU, who would further get  $Ring_A$  based on  $dpk_A$ , verify the signature  $\sigma_A$  by using  $Verify\_LRS(msg_{lvRng}, Ring_A, \sigma_A)$ , and finally remove  $dpk_A$  from the ring maintained by the RSU if the verification passes.

In case of malicious IoV client entity that refuses to send leave-ring request, one RSU should periodically broadcast existence query to all IoV client entities in the ring. If one IoV client fails to answer the query, then it will be removed from the ring maintained by the RSU.

## V. SECURITY ANALYSIS

In this section, we discuss how we design and implement our scheme to satisfy the IoV/IVTS authentication security requirements mentioned in Section IV-A. The security

characteristics of our scheme are mainly based on the security of the adopted signature/KEM algorithms and the Hyperledger Fabric system. The details are listed as follows.

Alg. 3: IoV client entity updates DPK in the RSU ring

**Prerequisites:**

**IoV client entity:**  $mpk_A, msk_A, dpk_{A_{rvk}}, dpk_{A_{updt}}, Cert_{dpk_{A_{rvk}}}, Cert_{dpk_{A_{updt}}}$   
**RSU:**  $\{dpk_1, \dots, dpk_{A_{rvk}}, \dots, dpk_i\} \rightarrow rsuRing$ ;

**function Update\_Ring()** in Client chaincode

```

  CertdpkArvk.serialNo  $\rightarrow sNo_1$ ;
  CertdpkAupdt.serialNo  $\rightarrow sNo_2$ ;
  "Request to update DPK in the RSU ring"  $\rightarrow urReq_A$ ;
  // The client entity makes a ring by using its revoked and updated
  DPKs and signs the update-ring request based on the update DPK
  {dpkArvk, dpkAupdt}  $\rightarrow Ring_A$ ;
  Sign_LRS(urReqA, RingA, dpkAupdt, (mpkA, mskA))  $\rightarrow \sigma_A$ ;
  // Invoke the function Update_Ring() in RSU chaincode
  shim.InvokeChaincode("RSU", urReqA,  $\sigma_A$ , sNo1, sNo2);

```

**function Update\_Ring()** in RSU chaincode

```

  // Upon receiving urReqA,  $\sigma_A$ , sNo1, sNo2 from Client
  chaincode
  RetrieveCertFromCA(sNo1)  $\rightarrow Cert_{dpk_{A_{rvk}}}$ ;
  RetrieveCertFromCA(sNo2)  $\rightarrow Cert_{dpk_{A_{updt}}}$ ;
  CertdpkArvk.dpk  $\rightarrow dpk_{A_{rvk}}$ ;
  CertdpkAupdt.dpk  $\rightarrow dpk_{A_{updt}}$ ;
  {dpkArvk, dpkAupdt}  $\rightarrow Ring_A$ ;
  if(Verify_LRS(urReqA, RingA,  $\sigma_A$ ) == 1)
  then UpdateRing(rsuRing, dpkArvk, dpkAupdt);

```

- 1) **Identity anonymity**. In our proposal, one IoV client entity (e.g., the vehicle OBU or pedestrian's mobile device) uses the anonymous CA-issued DPK certificate (i.e.,  $Cert_{dpk}$ ) to show its anonymized identity during the communication with other IoV client entities. The real-identity-based MPK certificate (i.e.,  $Cert_{mpk}$ ) is used only when the IoV client entity communicates with the CA and RSU. Therefore, no one (except CA and RSU) could know the real identity of the IoV client entity's DPK through intercepting the transmitted messages. Hence, our authentication scheme could preserve conditional identity privacy of IoV client entity.
- 2) **Identity traceability**. In our authentication scheme, each communication session between two IoV entities firstly needs a verification request of the related anonymous identity certificates sent onchain via RSU to CA, and all the identity-related keys and certificates are managed onchain by the CA, so the on-chain CA peer could easily trace the communication of each IoV client entity based on the on-chain certificates and reveal its real identity if necessary.
- 3) **Identity (un)linkability**. As mentioned in the 1) Identity anonymity, no IoV client entity can link a vehicle's real identity  $Cert_{mpk}$  to the vehicle's pseudonyms  $Cert_{dpk}$ . At the same time, our scheme supports linking multiple DPKs  $dpk_1, \dots, dpk_n$  generated from one same MPK  $mpk$  (i.e., one IoV client entity) by verifying the messages  $M_1, \dots, M_n$  and signatures  $\sigma_1, \dots, \sigma_n$  signed based on  $dpk_1, \dots, dpk_n$ , which is  $Link(m_i, R_i, \sigma_i, m_j, R_j, \sigma_j)$  where  $i, j \in [1, n]$ ,  $dpk_i \in R_i$  and  $dpk_j \in R_j$ . In this way, multiple



anonymous DPK certificates  $Cert_{mpk_1}, \dots, Cert_{mpk_n}$  generated from the same IoV client entity could be linked to prevent Sybil attack.

- 4) **Key exchange and message authentication.** Based on successful verification of identity and KEM certificates, two IoV entities could perform key exchange utilizing the KEM mechanism introduced in Section III-A and then get a negotiated session key  $sessKey$ . After successful key exchange, two IoV entities could perform efficient off-chain communication by using the session key  $sessKey$  as the symmetric encryption/decryption (e.g., AES) key or the Message Authentication Code (MAC) key to protect the confidentiality, integrity and authentication of the transmitted message.
- 5) **Post-quantum authentication.** All the cryptographic algorithms (including AES encryption/decryption, hash function, lattice-based linkable ring signature and NIST winner post-quantum signature/KEM algorithms) used in our authentication scheme are post-quantum, so our authentication scheme is resistant to quantum attacks.
- 6) **Resilience to other attacks.** Other attacks that our authentication scheme can resist are also listed as follows.

- **Man-in-the-middle attacks.** Under the mechanism of certificates and signature/MAC verification, the attacker cannot perform man-in-the-middle attacks as it does not have the sender's private key or session key to forge a valid signature or MAC.
- **Message modification attacks.** Each transmitted (e.g., P2P or broadcast) message  $M$  is attached with either one signature or one MAC. If the attacker modifies the transmitted  $M$  to  $M'$ , then  $M'$  will be discovered and discarded because the attacker cannot forge a valid signature or MAC for  $M'$  without the sender's private key or the session key.
- **Replay attacks.** For the sake of presentation simplicity, we ignore the global clock maintained the on-chain CA peer, which could provide a system-wide timestamp for each IoV entity and transmitted message. The timestamp embedded in each signature and MAC can keep the message freshness.
- **On-chain data modification attacks.** To protect the integrity of on-chain certificates and rings, we check the identity of the CA/RSU chaincode invoker by using the `getCreator()` API (in the *shim* package) in order to make sure that only the CA/RSU peer administrators can invoke the CA/RSU chaincode to manage the on-chain certificates and rings.

## VI. EVALUATION

In this section, we perform detailed performance evaluation of each on-chain API execution time, the off-chain communication time and the storage requirements. To further evaluate the feasibility of our authentication scheme in the IoV/IVTS environment, we show the effectiveness of our proposal in a blockchain-based simulation study.

TABLE II  
PQ ALGORITHM SETS BASED ON NIST PQ SIGNATURE AND KEM ALGORITHMS AT DIFFERENT SECURITY LEVELS

Sec. Levels	Alg. Set	Sig. Alg. Names	KEM Alg. Names
Level 1~2	Alg. Set 1/I	Dilithium2-AES	
	Alg. Set 2/II	Falcon-512	Kyber512-90s/HQC-128
	Alg. Set 3/III	SPHINCS+-SHA 256-128s-simple	
Level 3	Alg. Set 4/IV	Dilithium3-AES	Kyber768-90s/HQC-192
	Alg. Set 5/V	SPHINCS+-SHA 256-192s-simple	
Level 4~5	Alg. Set 6/VI	Dilithium5-AES	
	Alg. Set 7/VII	Falcon-1024	Kyber1024-90s/HQC-256
	Alg. Set 8/VIII	SPHINCS+-SHA 256-256s-simple	

The evaluation network topology is also shown in Figure 1, which includes one CA peer, 3 RSU peers and multiple IoV client entities connected to each RSU. As the experiment and simulation setup, we consider that the CA and RSU peers could be more powerful than the IoV client entities, so the tasks of CA and RSU are executed on VMs on local server, and each VM is based on Ubuntu 20.04 configured with 6 CPU cores of Intel Xeon E5-2670 throttled to 2.30GHz and 8GB memory. In contrast, to realize the OBU and pedestrian's mobile device functionalities, all the IoV client entities' tasks are executed on ARM Cortex-A53 CPU @ 1.4GHz with RAM 1 GB and Android 5.1. The blockchain system is built on Hyperledger Fabric 2.4.6, and the chaincodes together with the client codes are written in Go (1.18.2). We utilize the latest liboqs 0.6.0 library [37] together with its Go wrapper [38] to generate PQ key pair, sign/verify the PQ signature and encapsulate/decapsulate the shared secret.

### A. PQ Algorithm Sets

To achieve different NIST security levels (i.e., level 1~5), all the NIST PQ algorithms offer different key lengths and parameter sets. Therefore, in order to evaluate the performance of our proposal under different security levels, we divide the NIST PQ algorithms into three groups, namely level 1~2, level 3, level 4~5, which separately indicate low, medium and high security levels. We list our selected NIST PQ algorithm names by security levels in Table II, where one NIST PQ signature algorithm and one NIST PQ KEM algorithm at the same security level form an algorithm set (i.e., Algorithm Set 1~8 and I~VIII separately based on Kyber and HQC) that we utilize to evaluate our scheme performance under the corresponding security level. We only choose HQC from the three candidate KEM algorithms because the huge-key-size Classic McEliece algorithm is not suitable for the storage-limited IoV/IVTS environment and BIKE does not support high security level.

### B. On-Chain Execution and Off-Chain Communication Time

Since different NIST PQ algorithms have different performance, we test the on-chain execution time of all the CA/RSU/Client chaincode APIs (based on all

TABLE III  
OFF-CHAIN TIME OF IoV P2P COMMUNICATION

Size of Msg.	AESenc+MAC	AESDec+MAC	Msg. Trans. Time
1/8/16/48KB	3/3/3/5 $\mu$ s	7/8/8/11 $\mu$ s	0.2/0.4/0.4/1.7ms
248KB/1MB	8/21 $\mu$ s	14/35 $\mu$ s	7.2/21.7ms

the standardized algorithms), which correspond to the authentication-related operations that one CA/RSU/client can perform in our proposal. Because the execution time only differs a little between the experiments based on Kyber and HQC, we only explain the experiment results based on Kyber and attach the HQC-based results in [39].

As shown in Figure 3-(a), most of the certificate generation and revocation time of CA peer is quite short, except the certificate generation time when using the PQ signature algorithm SPHINCS+. The reason is that the private key of SPHINCS+ is much longer than other PQ signature algorithms and makes the private key chain-uploading process much longer. In Figure 3-(b), we can see that it takes around 100ms for the RSU peer to perform encapsulation/decapsulation/ring initialization operations and 200~400ms to generate its KEM certificate, but it is time-consuming for the RSU peer to manage the ring (e.g., adding one new DPK of the IoV client entity into the ring, updating the DPK of one IoV client entity in the ring and removing the DPK of one IoV client from the ring). This is because the size of the ring maintained by the RSU onchain could be big, which makes one read-write session of the on-chain ring time-consuming. In Figure 3-(c), one IoV client entity needs dozens of milliseconds to perform encapsulation/decapsulation/DPK verification operations, but it needs 300ms~800ms to join/update/leave the ring maintained by the RSU onchain due to the slow RSU ring management operations. It also takes much time for one IoV client entity to link two messages signed by other client entities because of the slow linking operation provided by the Kyber-based linkable ring signature algorithm. At last, the IoV client entity needs to upload/download the ring, which is used to sign/verify the broadcast message during broadcasting. The uploading/downloading operation is denoted as BroadcastInit/RecBroadcastInit in Figure 3-(c) and is fast.

With successful key exchange, two IoV entities could perform P2P communication with each other. We test the consumed time of this P2P communication time, which includes AES encryption/MAC protection time for the sent message, AES decryption/MAC recovery time for the received message and the protected message transmission time. As shown in Table III, the protection/recovery time could be ignored, and the message transmission time is also quite short even for 1MB transmitted message.

### C. Storage Requirements

We calculate the on-board and on-chain storage of our proposal, which are separately shown in Table IV and Table V. As shown in Table IV, one IoV client entity needs to store its KEM private key, one session key (shared with another client) and other keys (including its master public key, derived public

TABLE IV  
THE ON-BOARD STORAGE REQUIREMENT (IN KB)

KEM Alg.	KEM $sk$	sessKey	Oth. Keys	Total Size
Kyber512/768/1024-90s	2.1/3.1/4.1	32B	15.9	18.7/20.0/21.5
HQC-128/192/256	3.0/5.9/9.5	64B	15.9	23.2/30.6/39.5

key and related key materials). As one may see, the KEM private key size varies with the different security-level KEM algorithms, and the number of session keys would grow if the IoV client entity performs key exchange with more other client entities at the same time.

Supposing there are  $i$  IoV client entities (i.e., OBU number in Table V) connecting to one RSU, the blockchain public ledger needs to store the root certificate of the CA, the RSU identity/KEM certificates, the  $i$  clients' MPK/DPK/KEM certificates and the ring (containing the DPKs of the  $i$  clients) maintained by the RSU. We calculate the certificate sizes based on different security-level NIST signature/KEM algorithms and summarize the results in Table V.

### D. Blockchain-Based Simulation

To further evaluate the feasibility of our authentication scheme in the IoV/IVTS environment, we set up the blockchain-based system with one VM as the CA peer, one VM as the RSU peer and ten Android platforms as the IoV client entities. We let no/2/4 IoV client entities keep broadcasting message in order to increase the burden of the whole system, and test the execution time of one IoV client entity's necessary operations during performing authentication and communication with other IoV entities. These operations include joining/updating/leaving the ring, verifying the DPK, encapsulation/decapsulation (to get a session key), off-chain sending/receiving P2P message, uploading/downloading the ring used during broadcast (denoted as BroadcastInit/RecBroadcastInit) and off-chain broadcasting message/receiving broadcast message.

Again, the simulation results are quite similar based on Kyber and HQC (and only differ a little in encapsulation/decapsulation operations), so we only show the Kyber-based results and attach the HQC-based results in [39]. All the simulation results are shown in Figure 4, where we can see that it needs hundreds of milliseconds for one IoV client to join/update/leave the RSU ring and the time is growing when more IoV client entities start to broadcast. Luckily, one IoV client entity does not join/update/leave the RSU ring quite often when it is connecting to one RSU, so it is acceptable to have these operations. The second line of Figure 4 shows that two IoV client entities can quickly encapsulate/decapsulate with each other and then perform fast off-chain P2P message sending/receiving, and the increase of the system burden only affects the performance a little. While at the third line of Figure 4, it indicates that uploading/downloading the ring used during broadcast (i.e., BroadcastInit/RecBroadcastInit) is quick but sending/receiving broadcast message is time-consuming, which is caused by the slow linkable ring signature signing/verifying algorithms used when one broadcast message

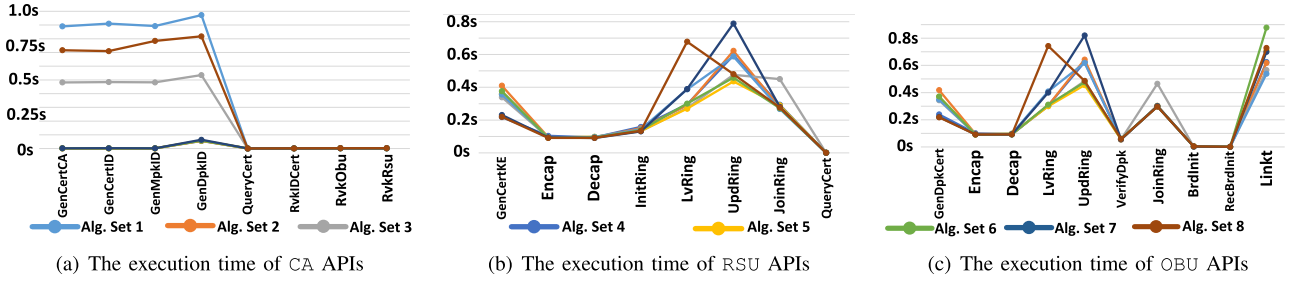


Fig. 3. The on-chain execution time of all the CA/RSU/Client chaincode APIs (under different security levels) corresponding to the CA/RSU/client operations in our authentication scheme.

TABLE V  
THE ON-CHAIN STORAGE REQUIREMENT OF OUR SCHEME (IN KB)

OBU Num.	Alg. Set 1/I	Alg. Set 2/II	Alg. Set 3/III	Alg. Set 4/IV	Alg. Set 5/V	Alg. Set 6/VI	Alg. Set 7/VII	Alg. Set 8/VIII
1	31.7/34.4	25.7/28.5	45.2/48.0	36.2/42.7	70.5/77.0	42.0/52.6	30.7/41.3	111.0/121.6
2	53.1/57.3	43.7/47.9	77.2/81.4	59.7/69.5	119.2/129.0	68.5/84.6	50.7/66.8	186.7/202.8
4	95.9/102.9	79.6/86.6	141.4/148.4	106.7/123.0	216.8/233.1	121.3/148.5	90.5/117.7	338.0/365.2
8	181.6/194.2	151.5/164.2	269.6/282.3	200.7/230.0	411.9/441.2	227.0/276.3	170.2/219.5	640.6/690.0

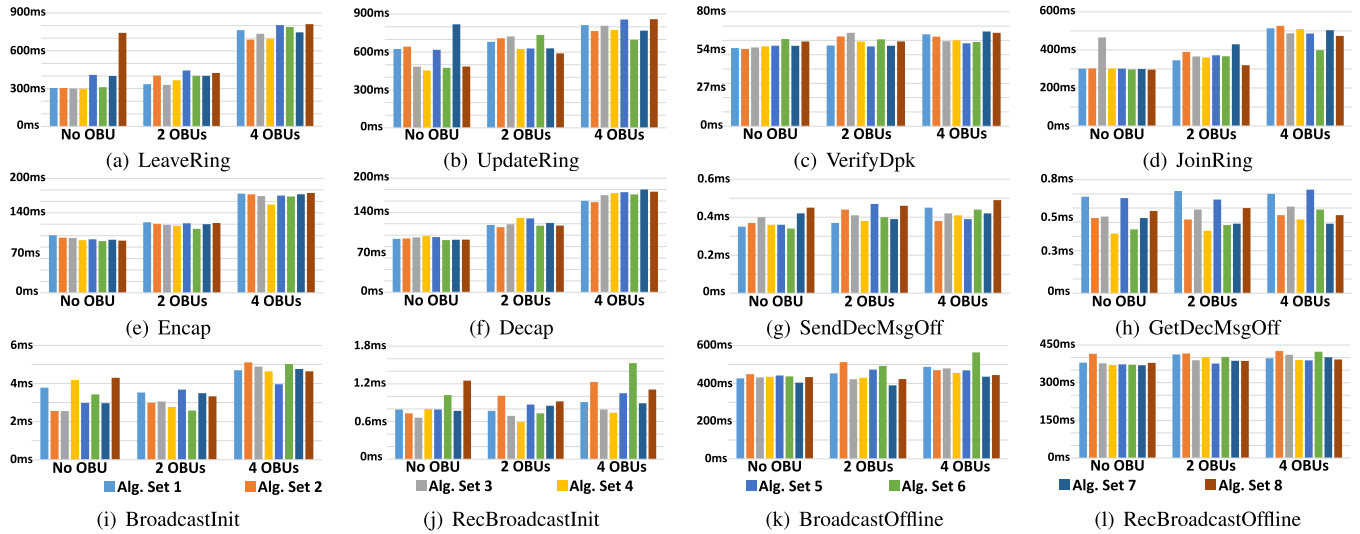


Fig. 4. The simulation results of our blockchain-based IoV authentication scheme.

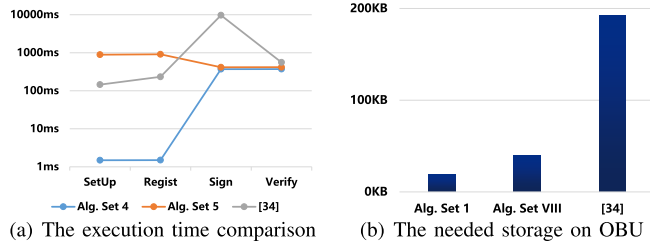


Fig. 5. Comparison of execution time and consumed size between our work and the most related research [34].

is broadcasted/received. To ease the broadcast-related performance problem, one IoV client entity could choose to broadcast and receive broadcast in two or more radio bands. In that case, two broadcast message could be sent and received per second at each radio band. Lastly, the performance of all the broadcast-related operations is hardly influenced by the increase of the system burden.

### E. Performance Comparison

Finally, we compare our scheme performance with the most related work [34] by using our most/least time-consuming algorithm sets (i.e., 4/5) and storage-consuming algorithm sets (i.e., 1/VIII). As shown in Figure 5(a), our scheme costs less time than [34] in every steps except the setup and registration phases (based on the most time-consuming algorithm set 5). However, the setup/registration phase needs to be executed much less times than the signature and verification operations, and the whole system performance would not be greatly affected. While on the other hand in Figure 5(b), our scheme needs much less OBU storage than [34] even based on the most storage-consuming algorithm set VIII and thus very friendly to the storage-limited OBUs.

## VII. CONCLUSION

To improve the security and privacy of vehicular communications in IoV/IVTS, we proposed anonymous, traceable

and linkable authentication scheme based on NIST PQ signature algorithms, the PQ linkable ring signature and Hyperledger Fabric. Our proposal preserved the identity privacy of IoV/IVTS's nodes and at the same time provided identity linkability in order to protect against Sybil attacks. We also developed post-quantum blockchain-based key exchange mechanism for IoV/IVTS's nodes based on NIST PQ KEM algorithm to help IoV/IVTS's nodes perform efficient message authentication encryption/decryption during P2P communication and broadcast. Finally, we performed detailed performance evaluation and blockchain-based simulation to show the feasibility of our scheme in the IoV-based IVTS environment. As the limitation of our scheme, it is better to perform the evaluation in the real IoV/IVTS system and thus we plan to deploy our scheme to the blockchain-based IVTS system with the real vehicles and RSUs (as the future research direction) in order to get the unsimulated experiment results and further optimize our scheme.

#### ACKNOWLEDGMENT

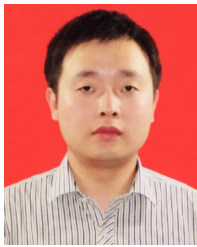
The authors would like to thank Bo Li and the experimental teaching center of the College of Informatics, Huazhong Agriculture University for providing the experimental environment and computing resources.

#### REFERENCES

- [1] M. Kamal, M. Tariq, G. Srivastava, and L. Malina, "Optimized security algorithms for intelligent and autonomous vehicular transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2038–2044, Feb. 2023.
- [2] P. M. Kumar, C. Konstantinou, S. Basheer, G. Manogaran, B. S. Rawal, and G. C. Babu, "Agreement-induced data verification model for securing vehicular communication in intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 980–989, Jan. 2023.
- [3] W. Yamany, N. Moustafa, and B. Turnbull, "OQFL: An optimized quantum-based federated learning framework for defending against adversarial attacks in intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 893–903, Jan. 2023.
- [4] M. Azees, P. Vijayakumar, and L. J. Deborah, "Comprehensive survey on security services in vehicular ad-hoc networks," *IET Intell. Transp. Syst.*, vol. 10, no. 6, pp. 379–388, 2016.
- [5] H. Peng, L. Liang, X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1064–1078, Feb. 2019.
- [6] J. Grover, V. Laxmi, and M. S. Gaur, "Sybil attack detection in VANET using neighbouring vehicles," *Int. J. Secur. Netw.*, vol. 9, no. 4, pp. 222–233, Dec. 2014.
- [7] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 1984, pp. 47–53.
- [8] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology—ASIACRYPT*, vol. 2894. Berlin, Germany: Springer, 2003, pp. 452–473.
- [9] Q. Mei, H. Xiong, Y. Zhao, and K.-H. Yeh, "Toward blockchain-enabled IoV with edge computing: Efficient and privacy-preserving vehicular communication and dynamic updating," in *Proc. IEEE Conf. Depend. Secure Comput. (DSC)*, Feb. 2021, pp. 1–8.
- [10] I. Ali, M. Gervais, E. Ahene, and F. Li, "A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs," *J. Syst. Archit.*, vol. 99, Oct. 2019, Art. no. 101636.
- [11] A. Elkhaili, J. Zhang, and R. Elhabob, "An efficient heterogeneous blockchain-based online/offline signcryption systems for Internet of Vehicles," *Cluster Comput.*, vol. 24, no. 3, pp. 2051–2068, Sep. 2021.
- [12] K. Prateek, F. Altaf, R. Amin, and S. Maity, "A privacy preserving authentication protocol using quantum computing for V2I authentication in vehicular ad hoc networks," *Secur. Commun. Netw.*, vol. 2022, pp. 1–17, Mar. 2022.
- [13] F. Qu, Z. Wu, F.-Y. Wang, and W. Cho, "A security and privacy review of VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 2985–2996, Dec. 2015.
- [14] Q. Mei, H. Xiong, Y.-C. Chen, and C.-M. Chen, "Blockchain-enabled privacy-preserving authentication mechanism for transportation CPS with cloud-edge computing," *IEEE Trans. Eng. Manag.*, pp. 1–12, Apr. 2022.
- [15] X. Zhang and C. Ye, "A novel privacy protection of permissioned blockchains with conditionally anonymous ring signature," *Cluster Comput.*, vol. 25, no. 2, pp. 1221–1235, Apr. 2022.
- [16] Y. Cao, S. Xu, X. Chen, Y. He, and S. Jiang, "A forward-secure and efficient authentication protocol through lattice-based group signature in VANETs scenarios," *Comput. Netw.*, vol. 214, Sep. 2022, Art. no. 109149.
- [17] J. Liu et al., "Lattice-based double-authentication-preventing ring signature for security and privacy in vehicular ad-hoc networks," *Tsinghua Sci. Technol.*, vol. 24, no. 5, pp. 575–584, 2019.
- [18] C. Jiao and X. Xiang, "Anti-quantum lattice-based ring signature scheme and applications in VANETs," *Entropy*, vol. 23, no. 10, p. 1364, Oct. 2021.
- [19] Q. Wu, J. Domingo-Ferrer, and Ú. Gonzalez-Nicolas, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications," *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 559–573, Feb. 2010.
- [20] NIST. (2022). *Post-quantum Cryptography—Selected Algorithms 2022*. [Online]. Available: <https://src.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>
- [21] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of Vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.
- [22] B. Mikavica and A. Kostić-Ljubisavljević, "Blockchain-based solutions for security, privacy, and trust management in vehicular networks: A survey," *J. Supercomput.*, vol. 77, no. 9, pp. 9520–9575, Feb. 2021.
- [23] D. Chhikara, S. Rana, G. Singh, D. Mishra, and N. Kumar, "Blockchain-based partial group key agreement protocol for intelligent transportation systems," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 16701–16710, Dec. 2023.
- [24] Z. Lu, Q. Wang, G. Qu, H. Zhang, and Z. Liu, "A blockchain-based privacy-preserving authentication scheme for VANETs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2792–2801, Dec. 2019.
- [25] L. Wang, D. Zheng, R. Guo, C. Hu, and C. Jing, "A blockchain-based privacy-preserving authentication scheme with anonymous identity in vehicular networks," *Int. J. Netw. Secur.*, vol. 22, no. 6, pp. 981–990, 2020.
- [26] S. Bao, A. Lei, H. Cruickshank, Z. Sun, P. Asuquo, and W. Hathal, "A pseudonym certificate management scheme based on blockchain for Internet of Vehicles," in *Proc. IEEE Intl. Conf. Dependable, Autonomous Secure Comput., Intl. Conf. Pervasive Intell. Comput., Intl. Conf. Cloud Big Data Comput., Intl. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCCom/CyberSciTech)*, Aug. 2019, pp. 28–35.
- [27] M. Khodaei and P. Papadimitratos, "Scalable & resilient vehicle-centric certificate revocation list distribution in vehicular communication systems," *IEEE Trans. Mobile Comput.*, vol. 20, no. 7, pp. 2473–2489, Jul. 2021.
- [28] X. Liu, H. Huang, F. Xiao, and Z. Ma, "A blockchain-based trust management with conditional privacy-preserving announcement scheme for VANETs," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4101–4112, May 2020.
- [29] H. Zhang, J. Liu, H. Zhao, P. Wang, and N. Kato, "Blockchain-based trust management for Internet of Vehicles," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1397–1409, Jul. 2021.
- [30] K.-A. Shim, "A survey on post-quantum public-key signature schemes for secure vehicular communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14025–14042, Sep. 2022.
- [31] P. S. L. M. Barreto, J. E. Ricardini, M. A. Simplicio, and H. K. Patil, "qSCMS: Post-quantum certificate provisioning process for V2X," *Cryptol. ePrint Arch.*, vol. 2018, p. 1247, Jan. 2018. [Online]. Available: <https://eprint.iacr.org/2018/1247>
- [32] S. Zhang, Y. Liu, Y. Xiao, and R. He, "A trust based adaptive privacy preserving authentication scheme for VANETs," *Veh. Commun.*, vol. 37, Oct. 2022, Art. no. 100516.
- [33] V. Srivastava, S. K. Debnath, B. Bera, A. K. Das, Y. Park, and P. Lorenz, "Blockchain-envisioned provably secure multivariate identity-based multi-signature scheme for Internet of Vehicles environment," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9853–9867, Sep. 2022.



- [34] D. S. Gupta, A. Karati, W. Saad, and D. B. da Costa, "Quantum-defended blockchain-assisted data authentication protocol for Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3255–3266, Mar. 2022.
- [35] Z. Liu, K. Nguyen, G. Yang, H. Wang, and D. S. Wong, "A lattice-based linkable ring signature supporting stealth addresses," in *Proc. 24th Eur. Symp. Res. Comput. Secur.* Berlin, Germany: Springer-Verlag, 2019, pp. 726–746.
- [36] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.* New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–15.
- [37] (2022). *Liboqs*. [Online]. Available: <https://github.com/open-quantum-safe/liboqs>
- [38] (2022). *Liboqs-Go: Go Bindings for Liboqs*. [Online]. Available: <https://github.com/open-quantum-safe/liboqs-go>
- [39] (2024). *Experiment Comparison Between Our and Related Works*. [Online]. Available: <https://github.com/BeIng5/PQ-ATL-AuthScheme>



**Shiwei Xu** received the B.S. degree in computer science and technology and the M.S. and Ph.D. degrees in information security from Wuhan University in 2006, 2008, and 2012, respectively. During the Ph.D. degree, he spent 18 months with the University of Birmingham. Currently, he is an Associate Professor in cyber security with the College of Informatics, Huazhong Agricultural University. He is also a member of CCF. His research interests include blockchain security, data privacy, and post-quantum cryptography.



**Tao Wang** received the B.E. degree in information security from Harbin Institute of Technology in 2019. He is currently pursuing the master's degree with the College of Informatics, Huazhong Agricultural University. His research interests include blockchain security and post-quantum cryptography.



**Ao Sun** received the bachelor's degree in information and computational science and the master's degree in computer software and theory from Huazhong Agricultural University in 2020 and 2023, respectively. He has joined Wuhan Maritime Communication Research Institute. His research interests include post-quantum cryptography and blockchain.



**Yan Tong** (Member, IEEE) received the B.S. and M.S. degrees in information security from Wuhan University in 2006 and 2008, respectively, and the joint Ph.D. degree from Wuhan University and City University of Hong Kong in 2012. Currently, he is an Associate Professor with the College of Science, Huazhong Agriculture University. His research interests include cryptography, blockchain, and information security.



**Zhengwei Ren** received the B.S. and Ph.D. degrees in information security from Wuhan University in 2009 and 2014, respectively. He is currently an Assistant Professor with Wuhan University of Science and Technology, Wuhan, China, where he is also an Adviser of Information Security Association. His research interests include applied cryptography and information security, with current focus on data security in cloud computing.



**Rongbo Zhu** (Member, IEEE) received the Ph.D. degree in communication and information systems from Shanghai Jiao Tong University, Shanghai, China, in 2006. From 2011 to 2012, he was a Visiting Scholar with the CNSR Laboratory, Virginia Tech, USA. He is currently a Professor with the College of Informatics, Huazhong Agricultural University, Wuhan, China. He has published over 60 articles in the areas of wireless networks and mobile computing.



**Houbing Herbert Song** (Fellow, IEEE) is currently a Professor with the University of Maryland, Baltimore County (UMBC), Baltimore, MD, USA. Prior to joining UMBC, he was a tenured Associate Professor in electrical engineering and computer science with Embry-Riddle Aeronautical University, Daytona Beach, FL, USA. He is an editor of eight books, the author of more than 100 articles, and an inventor of two patents. He also received more than ten Best Paper Awards from major international conferences, including IEEE CPSCOM-2019, IEEE ICII 2019, IEEE/AIAA ICNS 2019, IEEE CBDCOM 2020, WASA 2020, AIAA/IEEE DASC 2021, IEEE GLOBECOM 2021, and IEEE INFOCOM 2022. His research interests include cyber-physical systems/the Internet of Things, cybersecurity and privacy, and AI/machine learning/big data analytics.