This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

BCPPA: A Blockchain-Based Conditional Privacy-Preserving Authentication Protocol for Vehicular Ad Hoc Networks

Chao Lin, Debiao He, *Member, IEEE*, Xinyi Huang, *Member, IEEE*, Neeraj Kumar, *Senior Member, IEEE*, and Kim-Kwang Raymond Choo, *Senior Member, IEEE*

***Abstract*— While Vehicular Ad-hoc Networks (VANETs) can potentially improve driver safety and traffic mangement efficiency (e.g. through timely sharing of traffic status among vehicles), security and privacy are two ongoing issues that need to be addressed. Hence, security solutions such as conditional privacy-preserving authentication (CPPA) protocols have been proposed. However, CPPA protocols are generally far from being ready for deployment in VANETs, for example due key/certificate management limitations in PKI-based protocols or intractable private key updating in ID-based protocols. Although serveral blockchain-based CPPA (BCPPA) protocols have been proposed to mitigiate these challenges, there still exist some intractabilities such as revoking private key, or frequent interactions, or requiring an idea hardware. Thus, in this paper, we are motivated to propose a novel BCPPA protocol without these existing issues. Specifically, we present a PKI-based solution (using a typical digital signature protocol, such as ECDSA) based on Ethereum (a public blockchain), which is designed to facilitate secure communication in VANETs. In other words, we combine the blockchain technology and a key derivation algorithm to realize an effective certificate management. This reduces the need for participating vehicles to store a large number of private keys. To reduce the verification time cost, our BCPPA suppotrs replacing ECDSA with modified ECDSA for batch verification or directly adopting other PKI-based signatures with batch verification. In addition to introducing the concrete design, we also present the security requirements that our BCPPA protocol can satisfy. We then implement BCPPA in the Ethereum test network (i.e. *Rinkeby*) and provide simulations using Vanet-MobiSim and NS-2 to show its feasibility (i.e. milliseconds).**

***Index Terms*— Vehicular ad hoc network (VANET), conditional privacy-preserving authentication (CPPA), key derivation algorithm, blockchain, smart contract.**

Manuscript received September 20, 2018; revised November 2, 2019, January 11, 2020, and May 7, 2020; accepted June 10, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61972294 and Grant 61932016. The work of Kim-Kwang Raymond Choo was supported in part by the Cloud Technology Endowed Professorship and in part by the National Science Foundation CREST under Grant HRD-1736209. The Associate Editor for this article was C. T. Chigan. *(Corresponding author: Debiao He.)*

Chao Lin and Debiao He are with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Peng Cheng Laboratory Cyberspace Security Research Center, Shenzhen 518000, China (e-mail: linchao91@qq.com; hedebiao@163.com).

Xinyi Huang is with the School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350117, China, and also with the Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350117, China (e-mail: xyhuang81@gmail.com).

Neeraj Kumar is with the Department of Computer Science and Engineering, Thapar University, Patiala 147004, India, also with the Department of Computer Science and Information Engineering, Asia University, Taizhong 100600, Taiwan, and also with King Abdul Aziz University, Jeddah, Saudi Arabia (e-mail: neeraj.kumar@thapar.edu).

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249 USA, and also with the Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: raymond.choo@fulbrightmail.org).

Digital Object Identifier 10.1109/TITS.2020.3002096

I. Introduction

VEHICULAR Ad Hoc Network (VANET) is a self-organized ad-hoc network, where vehicles and roadside units (RSUs) are connected typically via wireless communications. Each participating vehicle is equipped with an On-Board Unit (OBU) (some wireless communication device), which provides the ability for vehicles to communicate with nearby vehicles and RSUs. The RSUs can further connect to the backbone network, for example via the Internet, for data sharing.

A typical VANET network model (see Figure 1) comprises Traffic Control Center (TCC), RSU, Vehicle, and Internet. There are three main modes of communications, namely: Wired/Wireless connection, Vehicle-to-Vehicle, and Vehicle-to-RSU. Wired/ wireless connection is used to connect vehicles and/or RSUs to the Internet, and the other two wireless communications are controlled by a Dedicated Short Range Communication (DSRC) protocol to facilitate short-range communication [1]. On basic of the OBUs and DSRC, vehicles can communicate with each other or with RSUs to share their current road traffic conditions (e.g. weather condition and congestion situation) or driving status (e.g. location and speed). This can help the vehicles to effectively avoid traffic congestions or possible traffic accidents by executing a timely response (e.g. re-routing to avoid traffic buildup) [2]. TCC can obtain these traffic messages from the RSUs via the Internet and take corresponding actions in a timely fashion (e.g. adjusting traffic lights).

Benefits of VANETs include supporting smart processing and real-time response in modern intelligent transportation systems. There are, however, potential safety concerns that should not be ignored especially during wireless communication mode, since wireless communication is more vulnerable

1524-9050 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

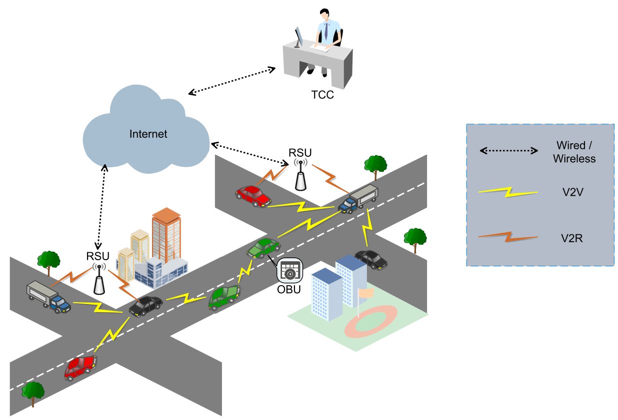


Fig. 1. A typical VANET network model.

than wired communication. For example, attackers can seek to create societal unrest by targeting such transportation system, for instance by intercepting, modifying, replay or deleting transmitting messages. Hence, the authenticity, validity and integrity of transmitted messages should be ensured to avoid impersonation or malicious modification. Successful attacks can result in real-world fatalities.

While message authentication can mitigate some of these attacks, we also need to consider protecting the privacy of vehicles (and their drivers/owners). For example, when a vehicle shares its traffic status with another RSU or vehicle, its identity will also be known. An attacker could mine such information and trace the route of the vehicle. Moreover, according to existing IEEE Standard [3], vehicles generally broadcast messages about their road traffic conditions and driving status periodically at an interval of 100-300 milliseconds. Such frequency in the broadcasted message facilitates the traceability of vehicles. Clearly, there are potential privacy and safety concerns.

One of the proposed solutions to support secure communications in VANETs is conditional privacy-preserving authentication (CPPA) [4], [5]. In the context of VANET, the vehicle's privacy should be conditionally protected in a CPPA protocol. This implies that the vehicle remains anonymous for most entities, although a trusted entity can extract the real identity of the vehicle. This allows one to find out a misbehaving vehicle (e.g. a vehicle who has sent a fabricated traffic status), so that appropriate penalty can be given to the offending vehicle.

Existing CPPA protocols for VANETs can be broadly categorized into PKI-based [4]–[6] and ID-based [7]–[10]. The latter category does not suffer from issues due to key/certificate preloading and revocation that exist in PKI-based protocols, and some schemes such as [1], [11], [12] further support batch verification to improve performance. However, these ID-based solutions result in new problems such as the intractability of revoking the vehicle's private key. This issue as well as other such as frequent interactions and requiring an idea hardware, are still existing in those newly raised blockchain-based CPPA (BCPPA) protocols (e.g. [13], [14]). Hence, we are motivated to propose an efficient PKI-based BCPPA protocol that eases the above issues.

A. Contributions

We demonstrate that Blockchain (a distributed ledger technology [15]–[17]) can be reliably used to store information (e.g. certificates or system parameters), which can then be retrieved by vehicles or RSUs to facilitate authentication. We also explain how smart contract can be used to establish the relationship among relevant information and perform revocation when the need arises. In addition, we introduce a key derivation algorithm to avoid the need of pre-storing a large number of keys in vehicle OBU. This really addresses the key escrow problem and guarantees the periodically updated private information, meaning that our proposal also relies on a realistic OBU.

We further propose a concrete BCPPA protocol using a typical digital signature scheme (e.g. ECDSA) in PKI systems. Our design supports the replacing of ECDSA with some modified ECDSA that supports batch verification (e.g. [18]–[20]) in order to minimize verification cost in VANETs. Note that other signatures with batch verification can also be integrated into our BCPPA protocol, which is of independent interests. Finally, we give the security and performance analysis to demonstrate the feasibility of our proposal.

B. Organization

We organize the rest of this paper as follows. Section II reviews existing CPPA protocols designed for VANETs. We introduce the blockchain-based system model and security requirements in Section III, prior to presenting the system building blocks in Section IV. We present our proposal and its security analysis in Sections V and VI, respectively. In Section VII, we implement our BCPPA in a Ethereum test network (i.e. *Rinkeby*[1]) with *MetaMask-Chrome*[2] and *Remix*,[3] and also provide two simulations using NS-2 for testing the average message delay and loss ratio. The findings are also presented in the section. Finally, we conclude this paper in Section VIII.

II. Related Work

The concept of CPPA was proposed by Raya and Hubaux [4] to address security and privacy concerns in VANETs. They also presented a concrete CPPA protocol using anonymous certificates, which can be realized using a modified PKI. That is, a large number of public/private key pairs and corresponding certificates are pre-loaded into vehicles' OBUs to achieve anonymous authentication (hiding the vehicle's real identity). When the vehicle wishes to share its traffic status, it should randomly choose a public/private key pair for message authentication via a signature. This will, however, result in significant storage costs (i.e. storing keys and certificates) for both vehicles and the relevant authority, as well as incurring significant cost to perform revocation of keys and certificates.

To mitigate the above deficiencies, Lu *et al.* [5] introduced a novel CPPA protocol via RSU-based anonymous certificates.

[1]https://www.rinkeby.io
[2]https://metamask.io/
[3]http://remix.ethereum.org

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

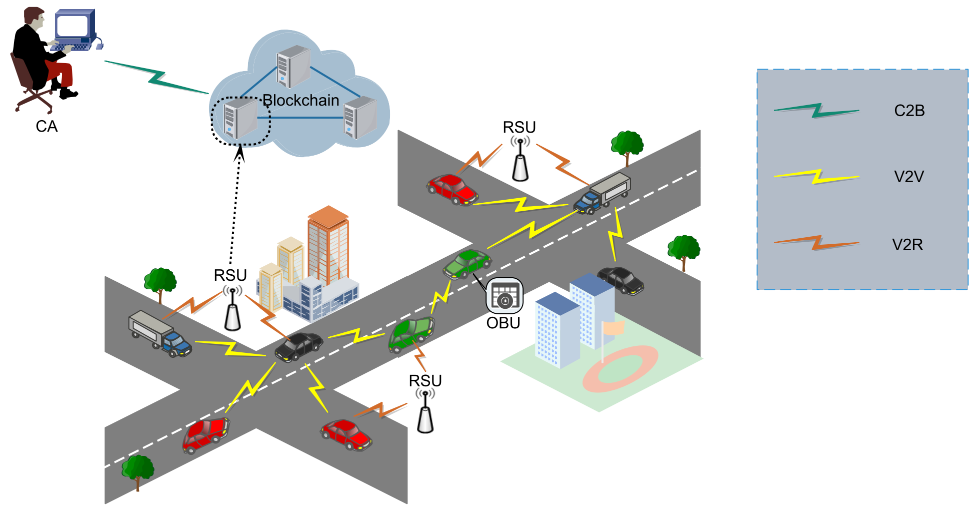


Fig. 2. Architecture of blockchain-based authentication protocol.

When the vehicle drives to an area near to a RSU, it will obtain a temporary anonymous certificate for authentication. Although one can achieve conditional privacy by frequently requesting for new anonymous certificates, signature signing and verification largely rely on online RSUs. This is inefficient in VANETs. Similarly, the CPPA protocols presented by Freudiger *et al.* [21] and Zhang *et al.* [6] incur significant storage cost for certificates in both vehicles and RSUs. In fact, one can observe from the literature a common limitation in existing CPPA protocols is key/certificate management complexity. Thus, there have been attempts to design ID-based CPPA protocols, such as those using ID-based signature [7], [22], [23], software-based solution [24], pseudo-ID-based solutions [1], and so on [8], [25]. All these protocols either focus on improving some existing solutions to achieve required security requirements or improving the efficiency of CPPA to support VANET applications.

However, most of these protocols either rely on an ideal hardware or are not suitable for multi-cloud environment. For solving the former challenge, Zhang *et al.* [11] proposed a Chinese Remainder Theorem-based CPPA Scheme and Zhang *et al.* [12] constructed a new CPPA scheme using multiple trusted authority one-time identity-based aggregate signature, both of which only require realistic tamper-proof devices. As the latter one, Cui *et al.* [26] designed a robust and extensible CPPA protocol that can meet the increasing diversified service needs in VANETs. Nevertheless, there still exists one common intractability of revoking vehicles' private keys in these ID-based solutions, which is an area that is relatively understudied.

Concurrently, there are several Blockchain-based CPPA (BCPPA) protocols have been proposed to solve those drawbacks existing in PKI-based solutions such as non-transparency of trusted authorities and heavy workload of revoking certificates. For example, Lu *et al.* [13] integrated blockchain and Merkle Patricia Tree to propose a novel BCPPA protocol with privacy protection and efficient certificate revocation, but it requires frequent interactions between vehicles and certificate authority to generate anonymous certificates. Zheng *et al.* [14] adopted pseudonym technology to design a ID-based BCPPA protocol with traceable anonymity, but which is faced with the requirement of ideal hardware and cannot resist against compromised certificate authority.

III. Problem Definition

In this section, we introduce the system model and the relevant security requirements.

A. System Model

The proposal BCPPA consists of four entities, i.e., **Certificate Authorities** (**CA**), **Road Side Uints** (**RSU**), **Vehicle** and **Blockchain Network** (**BN**) (see Figure 2), which connect with each other via the communications (C2B, V2R, and V2V). Here, C2B refers to the communication between CA and blockchain nodes (e.g. RSUs) that CA publishes transactions into the blockchain, V2R refers to that vehicles can request transaction data from the blockchain maintained by nearby RSUs, and V2V refers to the communication among vehicles via DSRC protocol. Note that the traffic control center and Internet in our system are consistent with that of the typical model in Figure 1, here we omit the description of them.

- **CA**: The CA is a trusted entity with enough resources (including computation and storage) who is responsible for managing certificates of vehicles' or RSUs' public keys. These certificates are signed by CA and embedded into the transactions via the C2B communication. In addition, CA builds the relationships between the issued public keys and its transaction identity using the smart contract such that one can conveniently retrieve the goal certificates from the blockchain. In our BCPPA, CA is the only entity who can obtain the real identity of the vehicle (i.e. conditional anonymity) from the intercepted messages.
- **RSU**: The RSU is a road side infrastructure which uses the DSRC protocol to communicate with OBUs. It also

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

serves as a full node (i.e. storing all the transaction data of the blockchain) which provides the APIs for retrieving transactions and triggering the chained smart contract (e.g. the test chain *rinkeby* of Ethereum).[4] Here, we assume that RSUs are fully trusted entities would not provide pseudo APIs.

- **Vehicle**: The Vehicle is equipped with an OBU which is an internal processing unit (with the tamper-proof property) can support DSRC protocol. Here, the OBU in our proposal is realistic, in the sense that the stored secrets inside can be periodically updated. Each OBU stores a private seed for deriving the vehicle's one-time private key via a key derivation algorithm, which efficiently avoids the storage of vast private keys. During the running process of the vehicle, the OBU regularly broadcasts its traffic status to nearby vehicles and RSUs. Here, the OBU mainly interacts with RSU for retrieving transactions via V2R communication and communicates with other vehicles via V2V communication.
- **BN**: The Blockchain Network provides the immutable, undeniable, and verifiable data storage forming as so-called transactions which constitute a blockchain. Concretely, we embed public certificates into the transaction such that the vehicles can obtain the goal certificates from the blockchain instead of preloading all the certificates in the OBUs. Here, we propose using a mature public blockchain (e.g. Ethereum) for our design that can be joined by anyone to maintain the blockchain. As mentioned above, RSUs join in this network as a full node supporting services (including retrieving transactions and triggering the smart contract) for nearby vehicles.

B. Security Requirements

In VANETs, security and privacy requirements are necessary to guarantee the secure communications among vehicles and RSUs. We investigate the existing research about authentication in VANETs such as [1], [4], [9] and the blockchain-based systems such as [27]–[29] to propose the following security requirements for a secure BCPPA protocol in VANETs.

1) **Message Authentication**: Vehicles can verify the authenticity of transmitted messages from other vehicles. It means that any modification on the message will be detected. Note that in our model, the C2B and V2R communications can be realized through HTTPs protocol (e.g. a web browser such as *rinkeby*), because RSUs mainly provide the blockchain data retrieval service for vehicles. Hence, we mainly consider this security requirement among V2V communications.
2) **Conditional Privacy Preservation**: The vehicle's privacy should be conditionally protected, meaning that only CA but other devices (e.g. RSUs and other vehicles) can extract the vehicle's real identity by analyzing the intercepted messages. In other words, RSUs and other vehicles can trust the transmitted messages are from some vehicles without knowing their real identities, which efficiently protects the privacy and security of vehicles. Once a vehicle broadcasts some inaccurate traffic statuses, they will be disclosed and revoked by CA.
3) **Unlinkability**: To prevent some malicious attackers from tracing the vehicle's travel path, two messages from the same vehicle cannot be linked.
4) **Birthday Collision Resilience**: The protocol should minimize the possibility of generating two same blocks simultaneously, i.e., it can efficiently resist birthday collision and avoid disputes between sub-blockchains.
5) **Hijacking Resilience**: The protocol should prevent attackers from hijacking transactions to realize a smooth transaction (i.e. ensuring the non-modifiability of transactions).
6) **51% Attack Resilience**: The protocol should prevent attackers from controlling majority of computing power (i.e. hashrate in PoW) which can directly reverse and alter past transactions to reach the double-spending target.
7) **Resilience to Other Attacks**: The blockchain-based CPPA protocol should be able to resist various common attacks (e.g. impersonation, modification, distributed denial of service, replay, man-in-the-middle, stolen verifier table, and side-channel attacks) in VANETs.

[4]https://www.rinkeby.io/

IV. System Building Blocks

A. Digital Signature

As mentioned in [4], a safety message in VANETs requires legitimacy but not confidentiality, because it does not contain any sensitive information. Hence, authentication is enough for the exchange of safety messages in VANETs and we adopt digital signatures (e.g. ECDSA) for the message authentication.

Assuming that each vehicle owns their public/private key pairs, they can digitally sign messages (denoted as Sign algorithm) using a private key such that the receiver can verify its authenticity using the corresponding public key (denoted as Verify algorithm). For authenticating the public key to a legitimate vehicle, a trusted authority (i.e. CA) is required to sign these public keys (generating certificates). This implies the use of Public Key Infrastructure (PKI).

In the typical PKI environment, if a vehicle (e.g. V_1) would like to send a safety message to other vehicles, it needs to sign the message by invoking Sign with its private key. Meanwhile, it should also provide the issued public key certificate from CA such that the receiver can verify the public key and then authenticate the message. That is, $(M, \mathsf{Sign}(sk_{V_1}, M, T), Cert_{V_1})$ is necessary for transmitting a safety message M, where sk_{V_1} is V_1's private key, T is the current timestamp for ensuring the message freshness, $Cert_{V_1}$ is the public key certificate of V_1.

In addition, the function of batch verification in digital signatures is an interesting property to reduce the verification time cost. Especially in the vast interactions of VANETs, this mechanism should be provided for the vehicles so that they can verify the validity of many messages simultaneously.

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Hence, some modified ECDSA schemes (such as [18]–[20]) supporting batch verification could be substituted directly in our proposal to achieve a lower verification cost. Here, we emphasize that any PKI-based signature with batch verification (e.g. Schnorr signature and Boneh–Lynn–Shacham (BLS) signature [30]) can also be integrated into our BCPPA for an improved performance.

Anyhow, the communication in our design does not transmit the CA's certificate, for which the certificates are all pre-recorded into the blockchain by CA for the direct retrieval by the vehicles. This can avoid the storage cost of storing abundant certificates in OBUs.

B. Key Derivation

In the current anonymous authentication protocols for VANETs based on certification (e.g. [4], [31]), a great deal of public/private key pairs and corresponding certificates should be pre-loaded into vehicles' OBUs. This causes a large storage space requirement of OBUs to store these key pairs and certificates. To avoid the necessity for preloading abundant key pairs, we propose using a key derivation algorithm (e.g. BIP32[5] widely used in Bitcoin) in our protocol. The security of BIP32 can be reduced to the discrete logarithm assumption, namely, knowledge of any child public keys pk_i alone is insufficient to recover the master public key pk_{root} or even other child public keys pk_j. This is the core for our proposal to achieve the property of anonymity and unlinkability [32].

Someone may argue that there exists a weakness in this algorithm that master private key sk_{root} can be recovered if given the master public key pk_{root} and any child private key sk_i. Here, we suggest using hardened keys for the account level in the tree[6] or adopting other improved key derivation algorithms such as [32] to mitigate this risk and achieve improved security. For a clearer understanding, we have drawn the flow chart (see Figure 3) for the key derivation algorithm with the following brief description.

- **Private type derivation**: This type is executed by the owner of private key (i.e. the OBUs equipped in the vehicles). A random seed is chosen to generate the root private key sk_{root} and chain code $chain_{root}$, which are used to derive a fresh private key sk_i for each communication. The corresponding $pk_{root} = sk_{root}G$ and $chain_{root}$ are transmitted to CA such that CA can derive the new public key pk_i and generate its certificate.
- **Public type derivation**: This type is executed by CA to derive the corresponding public key using the public information (i.e. pk_{root} and $chain_{root}$). The process is similar to that of private type derivation, which finally generates pk_i and $chain_i$. Here, we can check that $pk_i = sk_i G$, meaning that the consistency can be ensured (with regard to the same index i) even the CA and vehicles do not interact with each other to perform key management. This functionality also means that the CA can trace the master public key pk_{root} of a given derived public key pk_i, and hence guarantee the accountability of our proposal.

[5]https://github.com/bitcoin/bips/wiki/Comments:BIP-0032

[6]https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

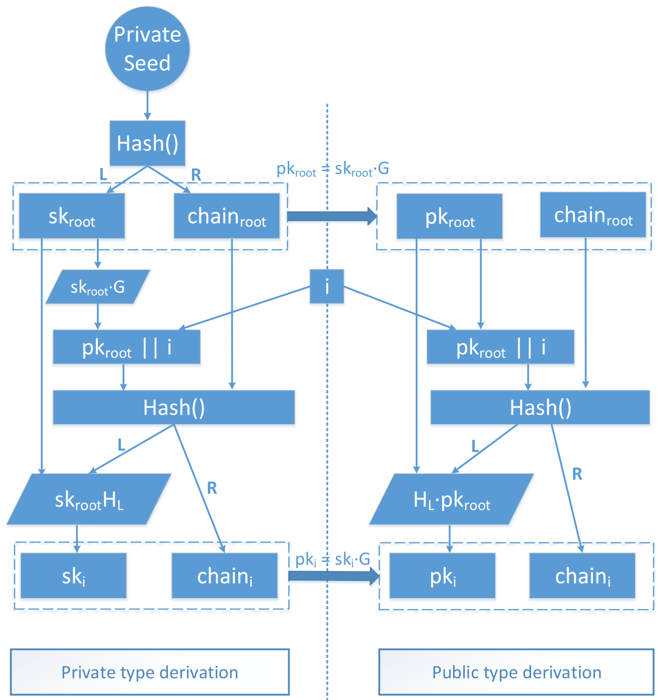


Fig. 3. The model of key derivation.

Note that CA should pre-issue the certificate of pk_i into the blockchain and update the indexes in the smart contract. This can guarantee that the vehicle can successfully retrieve the certificate corresponding to its new deriving private key.

C. Transaction

For the certificates issue into blockchain, we adopt embedding the certificate into the transaction. Specifically, the Ethereum (a public blockchain) is used in our design and its transaction format is reviewed as follows.

- *nonce*: This field records the account of transactions that a user has published, which is sequentially incremented for every transaction.
- *gasPrice*: This field defines the number of Wei (a unit of measurement in Ethereum) that per *gas* can be worth.
- *gasLimit*: This field defines the limitation number of *gas* used in the transaction.
- *to*: This field is padded with the receiver address for receiving tokens or triggering smart contract (when it is an address of a smart contract).
- *value*: This field represents the amount of tokens sent from a sender to a receiver.
- (v, r, s): This field is a ECDSA signature for authenticating the transaction information and its sender.
- *init*: This is one of optional fields that will be padded with EVM-code when the type of a transaction is smart contract creation (i.e. when the field *to* is $\varnothing$).
- *data*: This is the other optional field which will be padded with some input data such as parameters for triggering the

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

smart contract. In our design, we use this field to store the certificate of the vehicle's public key.

Note that a smart contract creation will return an address for the subsequent triggering of this contract. There are two kinds of transaction for triggering the contract, that is, "*eth_calls*" (executed by the local node) and "internal transactions"(invoked among different smart contracts). The details of designing our smart contract will be introduced in Section IV-E.

D. Blockchain

Blockchain, as the nucleus of Bitcoin's architecture, has attracted a lot of attention with a significant growth in both horizontal expansion (e.g. Bitcoin [33], Ethereum [34]) and vertical development (e.g. Hyperledger [35]). The former is called a public blockchain (i.e. anyone can join or quit the system to commonly maintain the blockchain), whereas the latter is only maintained by some trusted nodes (hence, it is called a permissioned or private blockchain). All these types of blockchain are maintained based on some consensus mechanisms (e.g. PoW [33] and PoS [36] in the public blockchain, PBFT [37] and RAFT [38] in the permissioned one) such that the blockchain is chronologically chained with immutability.

As mentioned above, the CA in our proposal needs to issue the certificates into the blockchain for others to retrieve. The smart contract function is necessary for mapping the public key to the transaction identity. Hence, we propose using the Ethereum (which has been widely used for designing DAPPs with *Solidity*[7] for writing smart contracts) for our design.

E. Smart Contract

Smart contracts are computerized transaction protocols that negotiate and perform a contract which obviates the need for a contractual clause [39]. It should be compiled into a piece of bytecode via a Turing complete language (e.g. *Solidity*), prior to being recorded into the chain forever. Then, its provided functions or application binary interfaces (ABIs) can be invoked via a transaction or a message from other contracts. It should be noted that each contract is a special account with its own address (named as smart contract address) and this address is indispensable for triggering the contract.

In our design, we mainly use the smart contract to map vehicles' public key to the transaction identities in the blockchain. The involved smart contract is simple but practical which only needs to provide the functions of update (only the CA can successfully invoked to map new transaction identity to the corresponding public key), get (can be invoked by anyone to obtain the transaction identity of a required public key), and deletetx (the same as update but for deleting the existing mapping when detecting malicious behaviors).

Here, the get is a *view* type function which is used for retrieving data from smart contract without any *gas* consumption and transaction confirmation. This can satisfy the low latency requirement of communications in VANETs. This smart contract also owns the function of certificate revocation, which is realized by deletetx algorithm. That is, only those valid and unrevoked certificates can be mapped in the contract, otherwise, the CA will delete the mapping to revoke the invalid and revoked certificates. The concrete contract design is briefly presented in Algorithm 1.

[7]http://solidity.readthedocs.io/en/v0.4.24/

Algorithm 1 Part 1 - Smart Contract on MapPkToTx

Require: Function name, invoked parameters
Ensure: *Setting up functions:*
address ca; *% Define the address of CA*
mapping (address → uint256) *public* $PK2TX$;
function MapPkToTx()
% Constructor, automatically invokes when this smart contract is deployed.
$ca = msg.sender$; *% Define the deployer as the CA*
function update(address $user$, uint256 $txid$) *public* returns (address $addr$)
% Invoked by CA to map a transaction identity to a public key.
require($msg.sender == ca$); *% Only the CA can successfully executed this algorithm*
$PK2TX[user] = txid$;
return $msg.sender$;
function get() *view* returns ($txid$)
% Invoked by any to retrieve a transaction identity to the required public key.
return $PK2TX[msg.sender]$;
function deletetx($target$) *public* returns ($txid$)
% Invoked by CA to delete the target mapping
require($msg.sender == ca$); *% Only the CA can successfully delete the existing mapping*
delete $PK2TX[target]$;

V. The Proposed BCPPA

In this section, we describe our BCPPA based on a public blockchain (i.e. Ethereum). Note that the digital signature scheme we adopt is ECDSA, however, this scheme could be replaced by some ones supporting batch verification (e.g. [18]–[20]) to reduce the verification time cost and achieve a more satisfactory performance for VANETs. No matter which algorithm will be used, the proposed BCPPA can efficiently support the secure V2V communication, which consists of three phases (as shown in Figure 4), i.e., **System Initialization Phase** (Step 1∼7), **Message Signing Phase** (Step 8∼12) and **Message Verification Phase** (Step 13∼15).

A. System Initialization Phase

This phase is executed by Vehicles and Certificate Authorities (CA) to initialize the key derivation and issue key certificates. Before the process of key certificates, all the vehicles should execute the private type derivation as shown in Fig. 3. That is, each of them randomly chooses a private seed (also named as a mnemonic word) to generate the private information (sk_{root} and $chain_{root}$). Then they compute the corresponding public information (pk_{root} and $chain_{root}$)

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

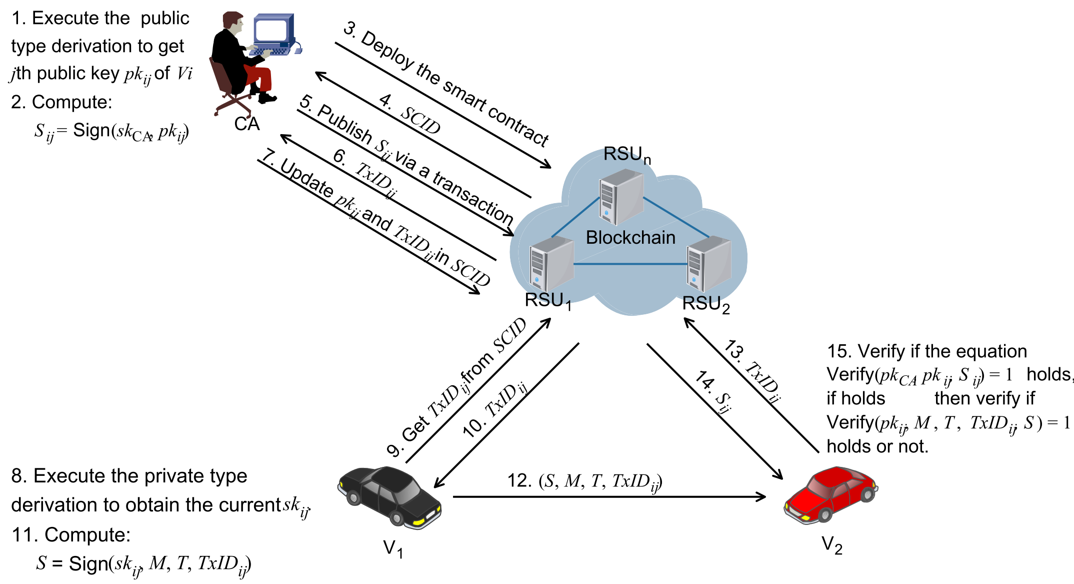


Fig. 4. The model of key derivation.

which are transmitted to CA. The former private information will be pre-loaded into the OBUs for deriving subsequent private keys by vehicles and the latter public ones are for deriving the corresponding public keys by CA.

For the convenient retrieval of certificates, CA deploy the smart contract to build the relationship between a public key and its relevant transaction identity. The obtained identity of smart contract (denoted by $SCID$) is also replied to all the vehicles for subsequently triggering this smart contract (i.e. require or update data in $SCID$). Then CA executes the following processes to issue key certificates in the blockchain for vehicles.

1) Assuming that the current serial number of vehicle V_i is j, CA executes the public type derivation to get the jth public key pk_{ij} of vehicle V_i.
2) Then it uses private key sk_{CA} to generate the certificate of pk_{ij} via computing $S_{ij} = \mathsf{Sign}(sk_{CA}, pk_{ij})$.
3) To record the certificate into the blockchain, CA embeds S_{ij} into a transaction that will be broadcast and chained into the blockchain by the miners. Then, the CA will obtain the transaction identity $TxID_{ij}$, which can be used to retrieve the certificate S_{ij}.
4) Finally, CA invokes the update algorithm to update pk_{ij} and $TxID_{ij}$ into the smart contract.

In addition, the CA could invoke the deletetx algorithm to revoke the compromised and expired vehicles. Once the mapping is deleted, the index of a certificate will no longer exist, meaning that this certificate has been revoked or invalided.

B. Message Signing Phase

This phase is executed by any vehicle to generate a message/signature pair for authenticating its identity and the message. This pair will be broadcast to nearby RSUs and vehicles via wireless communications such that all the vehicle can share their current traffic status with each other. Here, we assume that a vehicle (e.g. V_i) would like broadcast a message M to nearby vehicles (e.g. V_j), it will perform the following steps.

1) Due to the OBUs equipped in vehicles do not preload all the private keys, V_i should first execute the private type derivation to obtain the current private key (denoted as sk_{ij}) and computes $pk_{ij} = sk_{ij}G$.
2) Then, V_i triggers the smart contract $SCID$ via invoking Get algorithm to get the transaction identity (i.e. $TxID_{ij}$) of the public certificate corresponding to pk_{ij}. If the certificate is not revoked and V_i will obtain the $TxID_{ij}$; otherwise, it will get a null value.
3) Finally, V_i invokes the signing algorithm to generate the signature of M and $TxID_{ij}$ using sk_{ij}, that is, $S = \mathsf{Sign}(sk_{ij}, M, T, TxID_{ij})$, where T is the current timestamp. Then, the message /signature pair $(S, M, T, TxID_{ij})$ will be sent to V_j.

C. Message Verification Phase

In this phase, the verifier (a vehicle or a RSU) will verify if the received message/signature pair valid or not. Once the received information is valid, it means that the verifier can believe the received traffic status and perform some actions (e.g. changing lanes) if need be. According to the above subsection, the vehicle V_j will receive $(S, M, TxID_{ij})$ from V_i. Then, it can check the validity of $(S, M, TxID_{ij})$ with the certificate of CA's public key and the blockchain data (i.e. V_i's certificate). The verification process is presented as follows.

1) V_j gets the transaction data of $TxID_{ij}$ from the blockchain (via ABIs provided by the nearby RSU). Then, V_j can obtain the certificate S_{ij} of V_i's public key pk_{ij} from this transaction data.
2) Then V_j uses the certificate of CA's public key pk_{CA} to check the validity of S_{ij}, that is, it estimates the equation

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

$\mathsf{Verify}(pk_{CA}, pk_{ij}, S_{ij}) = 1$ holds or not. If not, V_j rejects this traffic status; otherwise, V_j uses the pk_{ij} to verify if the equation $\mathsf{Verify}(pk_{ij}, M, T, TxID_{ij}, S) = 1$ holds or not. If it holds, the message M is valid and authenticated from V_i.

VI. Security Analysis

In this section, we discuss the security requirements that our proposal can satisfy. That is mainly based on the security of the adopted digital signature scheme and the blockchain system. The details are given as follows.

1) **Message Authentication**: Due to the security of our adopted signature scheme (e.g. ECDSA), there exist no probabilistic polynomial time adversary can forge a valid message without the signing private key. In addition, the certificate signed by the CA can help the receiver to authenticate the sender's public key. Therefore, the receiver can verify the authenticity and integrity of the message $(S, M, T, TxID_{ij})$ through checking if both the equations $\mathsf{Verify}(pk_{CA}, pk_{ij}, S_{ij}) = 1$ and $\mathsf{Verify}(pk_{ij}, M, T, TxID_{ij}, S) = 1$ hold.
2) **Conditional Privacy Preservation**: In our proposal, the vehicle uses vast one-time public/private key pairs derived by a key derivation algorithm (which is hard to reverse the root pk_{root} and sk_{root} with those derived public keys). Note that the CA owns $(pk_{root}, chain_{root})$ and hence it can record the history of the derived public keys in the local database for relating some one-time public keys to the root pk_{root} (i.e. finding out the vehicle's real identity). It means that no one (except CA) can know the real identity of these one-time public keys through intercepting the transmitted messages. Hence, our proposal satisfies the aforementioned conditional privacy preservation.
3) **Unlinkability**: To broadcast a message M, the vehicle will derive a new private key and then signs M. To link two messages to the same senders, one should own the derivation ability to verify if one public key is derived from another one. However, the derivation process requires a chain code (i.e., $chain_i$) which is secretly keep by the CA. This represents that our proposal can reach to this security requirement.
4) **Birthday Collision Resilience**: This property is ensured because of the consensus mechanisms used in the Ethereum (i.e. PoW and PoS). These consensus mechanisms are used to combat forks and hence effectively decrease probability of blocks' birthday collisions.
5) **Hijacking Resilience**: All the transactions in Ethereum are signed by a digital signature scheme (i.e. ECDSA). This can resist hijacking attacks, because the security of ECDSA guarantees that no probabilistic polynomial time adversary can tamper the message of a transaction without invalidating the signatures.
6) **51% Attack Resilience**: To resist this attack, the only feasible measure is to make the cost of executing it as high as possible. For example, a higher issuance rate or a higher market price will help with that. The adopted Ethereum in our proposal uses a novel PoW with the "ASIC-resistant" expected to reduce economic incentives for mining centralization and then mitigates this risk.
7) **Resilience to Other Attacks**: Other attacks our proposal can resist are also listed as follows.
   - **Impersonation Attack**: To impersonate a legitimate vehicle to other vehicles, the attacker must generate a valid signature for its targeted message. However, this is not possible for any probabilistic polynomial time attacker according to the mentioned discussion and the receiver can detect this malicious attack by the simply verifying the signature. Hence, our BCPPA can resist the impersonation attack.
   - **Modification Attacks**: Assuming that an attacker modifies the broadcast message M', it will be discovered and discarded because it cannot forge a valid signature for M' without the sender's private key and the verification of the modified message /signature will return false.
   - **Distributed Denial of Service (DDoS) Attack**: Our BCPPA is benefited from the adopted Ethereum, among which DDoS requires a economically expensive transactions fees or gas consumptions. That's one of the attractive features because a server responds to your request for free on the regular Internet whereas the blockchain requires you to pay a price (which is actually huge).
   - **Replay Attack**: A fresh one-time private /public key pair is derived (by the vehicle and CA respectively) for the signing/verification of each communication. In addition, the timestamp embedded in each signature can also keep the message freshness. This can facilitate the vehicles in detecting any replay attack.
   - **Man-in-the-middle Attack**: From the above analysis of message authentication, it is clear that BCPPA provides secure authentication among the vehicles. Thus, it can also withstand this type of attacks.
   - **Stolen Verifier Table Attack**: The authentication in our design is based on the digital signatures without the need of maintaining a verifier table in Vehicles. Hence, the adversary cannot steal any verifier table for malicious attacks.

Side-channel

   - **Attacks**: In our BCPPA, only the secrets sk_{root} and $chain_{root}$ are stored in OBUs. These information are periodically updated, and hence it is much harder for an attacker to recover these secrets via launching side-channel attacks than to recover some unchanged secret embedded in existing ID-based solutions such as [1], [9]. As a matter of fact, most of existing secure protocols supporting online authentication have to embed similar secrets in the OBUs. This means that our BCPPA can achieve the similar security level of the secrets to these protocols. Furthermore, we suggest adopting multiplicative secret sharing MSS) technique [12], [40]

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

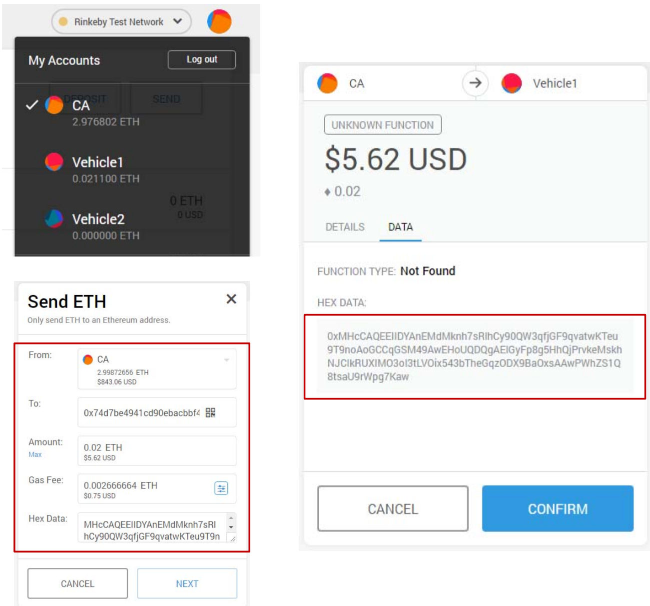


Fig. 5. The issue of certification.

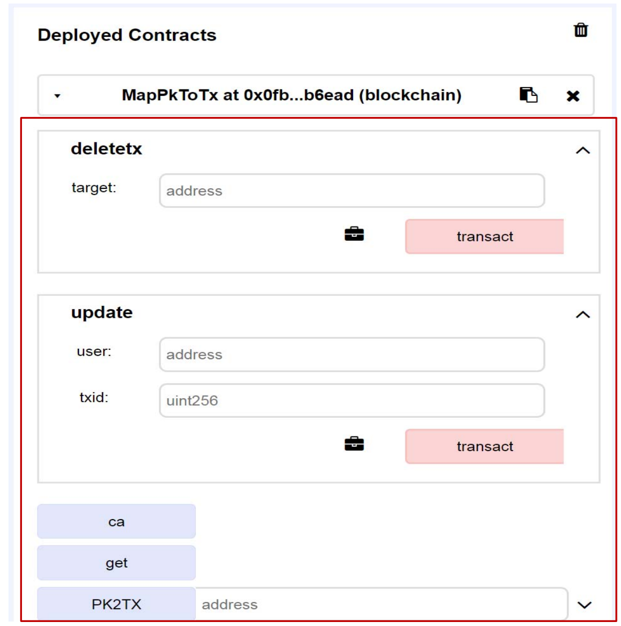


Fig. 6. The deployment of smart contract.

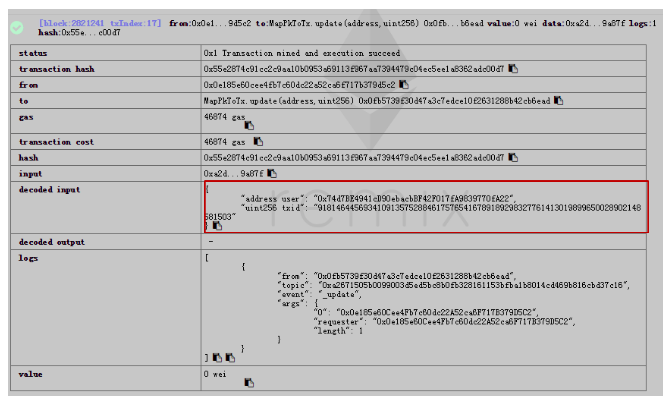


Fig. 7. Executing the update function.

to protect these secrets and increase the difficulty of launching powerful side-channel attacks.

VII. Performance Analysis

A. Implementation and Gas Cost

To discuss the feasibility of our BCPPA, we implemented it on *Rinkeby*[8] (a Ethereum test network). Here, *Rinkeby* not only provides a free request of funds, but also designs a user friendly web interface for a convenient block explorer. Moreover, we adopted a plug-in of Google Chrome (i.e. *MetaMask-Chrome*[9]) to connect *Rinkeby* in the Chrome and *Remix*[10] to deploy and invoke the smart contract. The details of this implementation are presented as follows.

1) Firstly, we used MetaMask to generate three accounts (CA, $Vehicle1$, and $Vehicle2$) for our test, addresses of which are 0×0e185e60Cee4Fb7c60dc22A52ca6F717B379D5C2, 0×74d7BE4941cD90ebacbBF42F017fA8397970fA22, and 0xe85bFDd5045dea3253092E50b6aFF7F124F7aC2b respectively. Then switched to the CA's account and requested 3 Ethers from the *Rinkeby* such that CA can publish transactions for issuing certificates. Here, we simulated the CA to issue the certificate of $Vehicle1$'s public key, that is, CA prepared the certificate and embedded it into a transaction. Once this transaction is recorded into the *Rinkeby*, a transaction identity would be returned such that others can retrieve it from the chain. The results are shown in Figure 5.
2) Then, we executed the followings as CA's identity. As shown in Figure 6, we deployed the smart contract into the *Rinkeby* using *Remix* and obtained its address (i.e. 0×0fb5739f30d47a3c7edce10f2631288b42cb6ead). We also invoked the update algorithm via *Remix* to update the $Vehicle1$'s public key with the aforementioned transaction identity into the Rinkeby (see Figure 7).
3) Next, we simulated the $Vehicle1$ to retrieve the location of its certificate chained in the *Rinkeby*. That is, we switched to the $Vehicle1$ account and invoked the get algorithm to obtain the information (see Figure 8). Here, the designed get is a *view* type algorithm which does not modify the state of the smart contract (hence, without any transaction confirmation time).
4) Finally, assuming that the $Vehicle2$ received a message from $Vehicle2$, it should retrieve the certificate from the *Rinkeby* according to the received transaction identity. Hence, we switched to the $Vehicle2$ account and get the targeted transaction in the *Rinkeby* (see Figure 9). Note that the transaction identity stored in the smart contract is decimal, which should be conversed into hexadecimal before being used to retrieve the transaction data.

[8]https://www.rinkeby.io
[9]chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#
[10]http://remix.ethereum.org

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

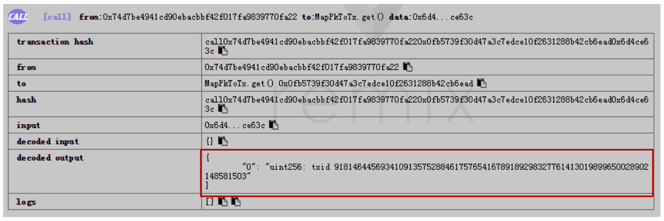


Fig. 8. Retrieving the transaction identity.

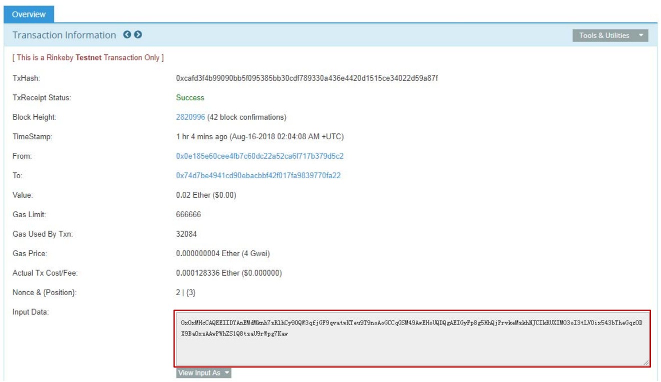


Fig. 9. Retrieving the transaction data.

TABLE I
SMART CONTRACT GAS COST (GAS PRICE = 2 GWEI, 1 ETHER = 188 USD)

Operation	Gas used	Actual cost (ether)	USD
deploy	408744	0.000817488	0.1536
update	68946	0.000137892	0.0259
get	24832	0.000049664	0.0093
deletetx	21588	0.000043176	0.0081

In addition, to test the cost in terms of transaction fees, we evaluated the gas cost of these operations (i.e. deploy update, get, and deletetx). From the result in Table I, the maximum cost was the deployment of smart contract (i.e. deploy) with approximately USD 0.1536, but which was only executed once. While all the other operations would be invoked repeatedly, the cost of them was less than USD 0.03 (especially the cost of get was about USD 0.0093). This means that one vehicle only needs to spend about USD 0.0093 for authenticating the other, which is an acceptable cost even the authentication is frequent.

B. Vehicle Authentication Efficiency

We also tested the time cost of key derivation KD algorithm, Sign and Verify algorithms of ECDSA, for which both the certificate issue and authentication phases involve these algorithms. The pairing-based library (version 0.5.12)[11] was used in our simulation and the adopted Type A pairings were constructed on the curve $y^2 = x^3+x$ over the field $\mathbb{F}_q$ for some primes q =3 mod 4. Each algorithm was executed 1000 times to obtain the average results. The concrete simulation platform is Ubuntu 16.04 (64 bits) with an Intel (R) Core (TM) i7-6700 CPU 3.40 GHZ and 3 GB RAM, and findings are shown in Table II.

[11]http://crypto.stanford.edu/pbc/

TABLE II
TIME COST (IN S) OF CRYPTOGRAPHIC ALGORITHMS

Algorithm	KD	Sign	Verify
Max Time	0.022427	0.008803	0.01264
Min Time	0.004111	0.003022	0.005471
Average Time	0.005061	0.003606	0.007184

Based on the test time of above algorithms, we finally evaluated the performance of our BCPPA from the perspectives of certificate management (maintained by the CA) and authentication in communications (among vehicles).

1) **Certificate Management**: As mentioned in [4], a vehicle should change its key within an interval of around 1 min. Assuming that a driver uses his/her car about average two hours per day and a driver requires about 43800 certificates per year. Because the solution in [4] requires that all the keys/certificates are generated at a time and pre-loaded into the OBUs. This not only causes an intolerable storage cost in OBUs, but also results in a crowded huge amount of computation cost in CA and a long time cost for checking the validity of certificate via the fast-growing CRL.
Our proposal can resolve these issues, where the vehicles do not need to pre-load the keys /certificates in OBUs but only the private seed and index (which can be used to derive a new private key in each new communication). In addition, the vehicles can obtain the certificates from *Rinkeby* directly.
As the certificate manager, the CA needs to pre-issue the certificates into the *Rinkeby*. Here, we propose that the CA can derive public keys for some day usages (e.g. about 240 for a interval of two days) at a time and generate the corresponding certificates. In addition, the certificate revocation can be directly realized via triggering the deletetx in the smart contract.
Although these operations also cause some computation and time costs (e.g. KD algorithm, Sign, and Transaction Confirmation), they have greatly reduced the complex computational cost compared to that of the existing PKI-based solutions. Here, we would not detail the concrete cost because these can be preprocessed by the CA. Instead, we focus on the follow analysis of the computation and time cost in a commutation.
2) **Authentication in Communications**: In each communication among two vehicles, it involves the message signing and verification. Hence we fist counted the operations and then computed the approximate time and communication costs, the comparative results of which are shown in Table III. Here, *eth_calls* represents the invocation of get algorithm from the smart contract and *transaction_retrieval* is the operation of retrieving transaction data from *Rinkeby*, $|TxID|$ is the length of a transaction hash (i.e. 32 bytes), $|S|$ is the length of a ECDSA signature (i.e. 64 bytes in our simulation),

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
COMPARISON WITH EXISTING PKI-BASED SOLUTIONS

Item	Our BCPPA		ECDSA-based protocols [41], [42]	
Time Cost (second)	Message Signing	*eth_calls* + Sign = 0.003606	Message Signing	Sign = 0.003606
	Message Verification	*transaction_retrieval* + 2 * Verify = 0.014368	Message Verification	2 * Verify = 0.014368
Communication Cost (byte)	$3\|TxID\|$+$2\|S\|$+$\|M\|$+$\|T\|$ = 264		$2\|S\|$+$\|M\|$+$\|T\|$ = 168	

$|M|$ is the length of a message (where we set as 32 bytes), and $|T|$ is the length of a timestamp (where we set as 8 bytes).

Both Message Signing and Verification phases of our BCPPA have the similar time costs to that of traditional ECDSA-based protocols [41], [42], for the time costs of *eth_calls* and *transaction_retrieval* can be omitted if without considering the transmission delay. In addition, the communication cost is 264 bytes in our BCPPA, which requires three additional hash values (i.e. 96 bytes) than protocols in [41], [42]. This cost is acceptable for our BCPPA owns some additional features (e.g. anonymity and traceability) than those traditional ECDSA-based protocols.

From the above discussion, we can find that the main time cost may be caused in the certificate management (which can be preprocessed) and the time cost of the authentication can reach to the millisecond level. This could satisfy with the feel-good experience requirement of users and demonstrates the major benefit of our BCPPA.

C. Message Authentication Delay and Loss Rate

To analyze the average message authentication delay and average message loss rate, we performed two simulations using VanetMobiSim[12] and NS-2[13] in a personal computer (Dell with Intel Core i7-6770 CPU 3.40 GHZ, 4 GB RAM and Ubuntu 16.04 OS). In our simulations,[14] the simulated scenario is in a map (see Fig. 10), which is split into four 0.5×0.5 km^2 blocks and every block is maintained by a RSU (with communication range of 600 m). The vehicles are equipped with average speed from 7.5 m/s to 40 m/s, and communication range of 300 m, as well as broadcast messages interval of 100 ms. The broadcast bandwidth bound was 6 Mbps, and the packet size was 264 bytes. Other parameters like Channel, Propagation, Phy, Mac, Queue, and Antenna were set as WirelessChannel, TwoRayGround, WirelessPhy, 802_11, DropTail/PriQueue, and OmniAntenna, respectively. The simulation time in each simulation was both 100 s.

According to the definitions of average packet delay (APD) and packet loss ratio (PLR) [43], together with simulators results, we obtains the final results as shown in Fig. 11 and Fig. 12. In the first simulator, we set the speed of vehicles as about 10-20 m/s with increasing the number of vehicles (i.e. density) from 5 to 100. From Figure 11, we observe that the

[12]http://vanet.eurecom.fr/

[13]https://www.isi.edu/nsnam/ns/

[14]Source codes in our simulation including smart contract, NS test code, VanetMobiSim test code are available at: https://github.com/colyn91/BCPPA

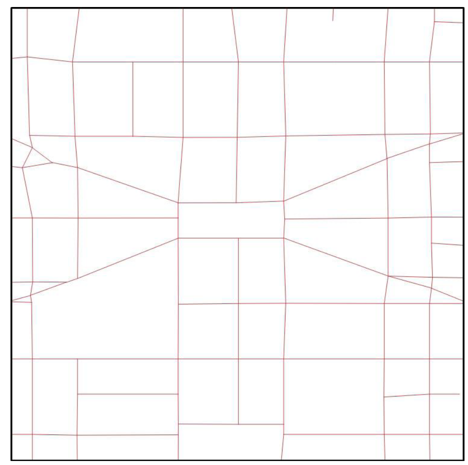


Fig. 10. Simulation scenario with 1×1 km^2.

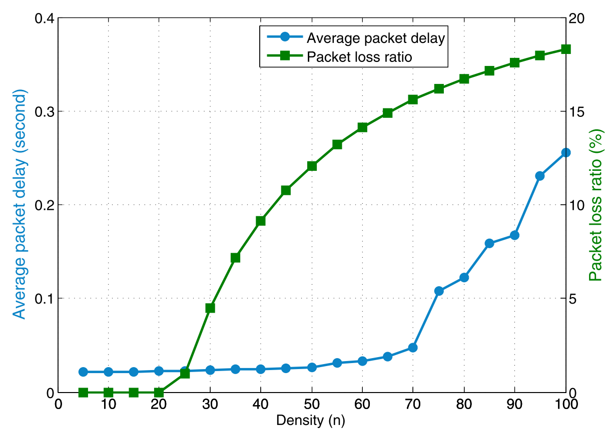


Fig. 11. The impact of density in packet delay and loss.

the APD is nearly unchanged (about 40 ms) when the density is less than 70, after that it grows rapidly. The PLR is almost zero at the beginning of the frame and sequentially increases when the density increases. Nevertheless, the increasing rate of PLR tends to be moderate after the density exceeds 70.

For all combinations of the above results, the performance degradation of VANETs will be caused when the density is more than 70. Thus, in the second simulator, we set the density as 70 to test the impact of different average speeds in the APD and PLR. The results are shown in Figure 12, on the one hand, the PLR keeps nearly constant even the average speed of vehicles increases. This means that the average speed of vehicles has little influence on the PLR in the same density, which corresponds to the reality that only those packets out of scope of vehicles will be lost. On the

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

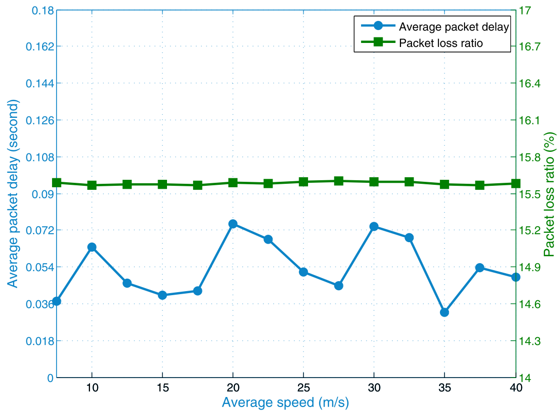


Fig. 12. The impact of speed in packet delay and loss.

other hand, the APD is fluctuating when the average speed changes. It may be caused by that the different average speeds of vehicles lead to the unpredictable distance change among vehicles and hence different APDs. Nevertheless, the span of APD is not more than 60 ms.

VIII. Conclusion

As driverless vehicles become more commonplace, VANETs will play an increasingly important role, for example in enhancing traffic safety and efficiency. In turn, this necessitates the design of secure and practical communication mechanism. Thus, in this paper, we presented a novel blockchain-based CPPA (BCPPA) protocol designed to facilitate secure communication in VANETs. Specifically, we integrated both blockchain and key derivation algorithm to design a novel BCPPA protocol. In our proposed BCPPA protocol, we use ECDSA as the building block which can also be replaced by some modified ECDSA (or any other PKI-based signature) with batch verification to improve the performance. We also demonstrated the security and utility of the proposed protocol.

Future research includes implementing the proposed mechanism in the authors' institutions with the aims of evaluating both security and performance in a real-world environment.

References

[1] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2681–2691, Dec. 2015.

[2] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Vehicular ad hoc networks: A new challenge for localization-based systems," *Comput. Commun.*, vol. 31, no. 12, pp. 2838–2849, Jul. 2008.

[3] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environment Security Services for Applications and Management Messages*, IEEE Standard 1609.2-2006, 2006.

[4] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Secur.*, vol. 15, no. 1, pp. 39–68, 2007.

[5] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. 27th Conf. Comput. Commun. (IEEE INFOCOM)*, Phoenix, AZ, USA, Apr. 2008, pp. 1229–1237.

[6] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, "RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks," in *Proc. IEEE Int. Conf. Commun.*, Beijing, China, May 2008, pp. 1451–1457.

[7] K.-A. Shim, "CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 4, pp. 1874–1883, May 2012.

[8] J. K. Liu, T. H. Yuen, M. H. Au, and W. Susilo, "Improvements on an authentication scheme for vehicular sensor networks," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2559–2564, Apr. 2014.

[9] M. Bayat, M. Barmshoory, M. Rahimi, and M. R. Aref, "A secure authentication scheme for VANETs with batch verification," *Wireless Netw.*, vol. 21, no. 5, pp. 1733–1743, Jul. 2015.

[10] T. W. Chim, S. M. Yiu, L. C. K. Hui, and V. O. K. Li, "SPECS: Secure and privacy enhancing communications schemes for VANETs," *Ad Hoc Netw.*, vol. 9, no. 2, pp. 189–203, Mar. 2011.

[11] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Trans. Dependable Secure Comput.*, early access, Mar. 11, 2019, doi: 10.1109/TDSC.2019.2904274.

[12] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu, "Distributed aggregate privacy-preserving authentication in VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 516–526, Mar. 2017.

[13] 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.

[14] D. Zheng, C. Jing, R. Guo, S. Gao, and L. Wang, "A traceable blockchain-based access authentication system with privacy preservation in VANETs," *IEEE Access*, vol. 7, pp. 117716–117726, 2019.

[15] M. Swan, *Blockchain: Blueprint for a New Economy*. Newton, MA, USA: O'Reilly Media, Inc., 2015.

[16] C. Lin, D. He, X. Huang, X. Xie, and K.-K.-R. Choo, "Blockchain-based system for secure outsourcing of bilinear pairings," *Inf. Sci.*, vol. 527, pp. 590–601, Jul. 2020.

[17] C. Lin, D. He, X. Huang, M. Khurram Khan, and K.-K.-R. Choo, "A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems," *IEEE Access*, vol. 6, pp. 28203–28212, 2018.

[18] S. Karati and A. Das, "Faster batch verification of standard ECDSA signatures using summation polynomials," in *Proc. 12th Int. Conf. Appl. Cryptography Netw. Secur. (ACNS)*, in Lecture Notes in Computer Science, vol. 8479, I. Boureanu, P. Owesarski, and S. Vaudenay, Eds. Lausanne, Switzerland: Springer, Jun. 2014, pp. 438–456.

[19] S. Karati, A. Das, and D. Roychoudhury, "Randomized batch verification of standard ECDSA signatures," in *Proc. 4th Int. Conf. Secur., Privacy, Appl. Cryptogr. Eng. (SPACE)*, in Lecture Notes in Computer Science, vol. 8804, R. S. Chakraborty, V. Matyas, and P. Schaumont, Eds. Pune, India: Springer, Oct. 2014, pp. 237–255.

[20] S. Karati, A. Das, D. R. Chowdhury, B. Bellur, D. Bhattacharya, and A. Iyer, "Batch verification of ECDSA signatures," in *Proc. 5th Int. Conf. Cryptol. Afr. Prog. Cryptol. (AFRICACRYPT)*, in Lecture Notes in Computer Science, vol. 7374, A. Mitrokotsa and S. Vaudenay, Eds. Ifran, Morocco: Springer, Jul. 2012, pp. 1–18.

[21] J. Freudiger, M. Raya, F. Mark, P. Papadimitratos, and J. P. Hubaux, "Mix-zones for location privacy in vehicular networks," in *Proc. WiN-ITS*, 2007, pp. 1–7.

[22] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. 27th Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2008, pp. 246–250.

[23] C. Zhang, P.-H. Ho, and J. Tapolcai, "On batch verification with group testing for vehicular communications," *Wireless Netw.*, vol. 17, no. 8, pp. 1851–1865, Nov. 2011.

[24] T. W. Chim, S. Yiu, L. C. K. Hui, Z. L. Jiang, and V. O. K. Li, "SPECS: Secure and privacy enhancing communications schemes for VANETs," in *Proc. 1st Int. Conf. Ad Hoc Netw. (ADHOCNETS)*, in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 28, J. Zheng, S. Mao, S. F. Midkiff, and H. Zhu, Eds. Niagara Falls, ON, Canada: Springer, Sep. 2009, pp. 160–175.

[25] J. Zhang, M. Xu, and L. Liu, "On the security of a secure batch verification with group testing for VANET," *Int. J. Netw. Secur.*, vol. 16, no. 5, pp. 355–362, 2014.

[26] J. Cui, X. Zhang, H. Zhong, J. Zhang, and L. Liu, "Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1654–1667, 2020, doi: 10.1109/TIFS.2019.2946933.

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[27] C. Lin, D. He, X. Huang, K.-K.-R. Choo, and A. V. Vasilakos, "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *J. Netw. Comput. Appl.*, vol. 116, pp. 42–52, Aug. 2018.

[28] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K.-K.-R. Choo, "HomeChain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 818–829, Feb. 2020.

[29] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 840–852, Sep. 2018.

[30] G. Fuchsbauer, M. Orrù, and Y. Seurin, "Aggregate cash systems: A cryptographic investigation of mimblewimble," in *Proc. 38th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EUROCRYPT)*, in Lecture Notes in Computer Science, vol. 11476, Y. Ishai and V. Rijmen, Eds. Darmstadt, Germany: Springer, May 2019, pp. 657–689.

[31] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho, and X. Shen, "Security in vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 88–95, Apr. 2008.

[32] G. Gutoski and D. Stebila, "Hierarchical deterministic bitcoin wallets that tolerate key leakage," in *Proc. 19th Int. Conf. Financial Cryptogr. Data Secur. (FC)*, in Lecture Notes in Computer Science, vol. 8975, R. Böhme and T. Okamoto, Eds. San Juan, PR, USA: Springer, Jan. 2015, pp. 497–504.

[33] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*. Consulted, 2008. [Online]. Available: https://git.dhimmel.com/bitcoin-whitepaper/

[34] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[35] E. Androulaki *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf. (EuroSys)*, R. Oliveira, P. Felber, and Y. C. Hu, Eds. New York, NY, USA: ACM, Apr. 2018, pp. 30:1–30:15.

[36] D. Larimer. (2013). *Transactions as Proof of Stake*. [Online]. Available: https://bravenewcoin.com/assets/Uploads/TransactionsAsProofOfStake10.pdf

[37] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proc. 3rd USENIX Symp. Operating Syst. Design Implement. (OSDI)*, M. I. Seltzer and P. J. Leach, Eds. New Orleans, LA, USA: USENIX Association, Feb. 1999, pp. 173–186.

[38] D. Ongaro and J. K. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, G. Gibson and N. Zeldovich, Eds. Philadelphia, PA, USA: USENIX Association, Jun. 2014, pp. 305–319.

[39] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1594–1605, Apr. 2018.

[40] X. Lai, J. Zhou, and H. Li, eds., *Information Security: 14th International Conference* (Lecture Notes in Computer Science), vol. 7001. Xi'an, China: Springer, Oct. 2011.

[41] S. S. Manvi, M. S. Kakkasageri, and D. G. Adiga, "Message authentication in vehicular ad hoc networks: ECDSA based approach," in *Proc. Int. Conf. Future Comput. Commun.*, Apr. 2009, pp. 16–20.

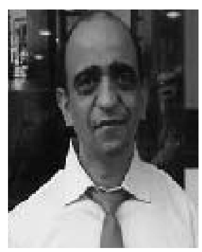
[42] K. Ravi and S. A. Kulkarni, "A secure message authentication scheme for VANET using ECDSA," in *Proc. 4th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2013, pp. 1–6.

[43] Y. Liu, L. Wang, and H.-H. Chen, "Message authentication using proxy vehicles in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 8, pp. 3697–3710, Aug. 2015.

Chao Lin received the bachelor's and master's degrees from the School of Mathematics and Computer Science, Fujian Normal University, in 2013 and 2017, respectively. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University. His research interests include authentication of graph data and blockchain security.

Debiao He (Member, IEEE) received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, in 2009. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. His main research interests include cryptography and information security, in particular, and cryptographic protocols.

Xinyi Huang (Member, IEEE) received the Ph.D. degree from the University of Wollongong, Australia. He is currently a Professor with the School of Mathematics and Computer Science, Fujian Normal University, China, and the Co-Director of the Fujian Provincial Key Laboratory of Network Security and Cryptology. He serves on the Editorial Board of the *International Journal of Information Security*, the program/general chair or program committee member in over 80 international conferences. His research interests include applied cryptography and network security. He is an Associate Editor of the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING.



Neeraj Kumar (Senior Member, IEEE) received the Ph.D. degree in CSE from Shri Mata Vaishno Devi University, Katra, India. He is currently an Associate Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. His research interests include mobile computing, parallel/distributed computing, multiagent systems, service oriented computing, routing and security issues in mobile ad-hoc, sensor, and mesh networks. He has more than 400 technical research articles in leading journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE Intelligent Transportation Systems Society, the IEEE TWPS, the IEEE SENSOR JOURNAL, the *IEEE Communications Magazine*, the *IEEE Walnut Creek Magazine*, the IEEE *Network Magazine*, and conferences. He has authored/edited 10 books from Elsevier, Springer, CRC, and BPB publications. His research was supported by DST, TCS, and UGC. He has guided many students leading to their M.E. and Ph.D., degrees.

Kim-Kwang Raymond Choo (Senior Member, IEEE) received the Ph.D. degree in information security from the Queensland University of Technology, Australia, in 2006. He currently holds the Cloud Technology Endowed Professorship, The University of Texas at San Antonio (UTSA). In 2016, he was named the Cybersecurity Educator of the Year-APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn). In 2015, he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is also a fellow of the Australian Computer Society. He was a recipient of the 2019 IEEE Technical Committee on Scalable Computing (TCSC) Award for Excellence in Scalable Computing (Middle Career Researcher), the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, the IEEE TrustCom 2018 Best Paper Award, the ESORICS 2015 Best Research Paper Award, the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, the Fulbright Scholarship in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award in 2008.

Authorized licensed use limited to: University of Texas at San Antonio. Downloaded on August 17,2020 at 17:53:51 UTC from IEEE Xplore. Restrictions apply.