Hindawi
Security and Communication Networks
Volume 2023, Article ID 9856810, 1 page
https://doi.org/10.1155/2023/9856810

Retraction

Retracted: GAP-MM: 5G-Enabled Real-Time Autonomous Vehicle Platoon Membership Management Based on Blockchain

Security and Communication Networks

Received 31 October 2023; Accepted 31 October 2023; Published 1 November 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] B. Wu, Q. Wu, and Z. Ying, "GAP-MM: 5G-Enabled Real-Time Autonomous Vehicle Platoon Membership Management Based on Blockchain," *Security and Communication Networks*, vol. 2022, Article ID 7567994, 14 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7567994, 14 pages
https://doi.org/10.1155/2022/7567994

Research Article

GAP-MM: 5G-Enabled Real-Time Autonomous Vehicle Platoon Membership Management Based on Blockchain

Bin Wu,[1] **Qilin Wu**,[1] **and Zuobin Ying**[2]

[1]*School of Information Engineering, Chaohu Univeristy, Chaohu 238000, China*
[2]*Faculty of Data Science, City University of Macau, Macau 999078, China*

Correspondence should be addressed to Bin Wu; 054013@chu.edu.cn

Received 29 October 2021; Revised 27 December 2021; Accepted 31 December 2021; Published 12 May 2022

Academic Editor: Jian Su

Autonomous Vehicle Platoon (AVP) is the most anticipated application of 5G ultrareliable and low latency communications. By joining the preexisting platoon to form the real-time AVP (RAVP), individual vehicles could gain benefits such as fuel consumption reduction and traffic safety enhancement. Unlike the scheduled AVP which only contains fixed vehicles, members in RAVP change frequently since individual vehicles would want to join or leave the platoon any time. Besides, malicious vehicles may attempt to sneak into the RAVP and try to manipulate the platoon. Public key cryptography and access control mechanisms can be adopted to create a relatively isolated area for communication inside the platoon. Nevertheless, there exist few works that take into account the regulation of the dynamic change members in RAVP. In this paper, we propose a membership management scheme for 5G-enabled RAVP by integrating revocable attribute-based encryption (RABE) and blockchain, namely, GAP-MM. It realizes fine-grained access control of key distribution and malicious vehicle's key revocation efficiently. The sufficient evaluations and security analysis indicate that GAP-MM is practical for RAVP scenario in terms of both efficiency and security.

1. Introduction

With the increasingly growing number of vehicles, traffic congestion has become a serious drawback, which affects the global economy, as it has been an indiscriminate phenomenon all around the world. According to the report from INRIX, in 2013, traffic congestion robbed the US economy of $124 billion. Without significant action to alleviate congestion, this cost is expected to increase by 50 percent to $186 billion by 2030. The cumulative cost over the 17-year period is projected to be $2.8 trillion, the same amount Americans collectively paid in US taxes last year [1]. Considering the hardness of rescheduling the road plan or increasing the basic traffic infrastructure, a novel transportation pattern has been studied to relieve the pressure of congestion, namely, autonomous vehicle platooning (AVP).

AVP is a driving pattern which allows vehicles to drive close together. Usually, there is a platoon leader (PL) in the AVP. The rest of the vehicles are recognized as platoon members (PMs). During the entire journey, the PL is responsible for collecting road information from the facilities (According to the different vehicular communication modes, the facilities could be either Road Side Unit (RSU) in WLAN or base station (e.g., gNB, eNodeB) in cellular network.) and other external vehicles. PMs only have to follow the instructions given by the PL. All AVP members can communicate with each other either by using Dedicated Short Range Communications (DSRC) or through Cellular Vehicle-to-Everything (C-V2X). However, it is not until the 5th-Generation (5G) that put forward the ultrareliable and low latency communications (uRLLC) that AVP became a reality. For instance, it would take about only 2.5 cm to apply brakes with 5G compared to 1.5 m with 4G for a vehicle. Thus, the space between front and following vehicles can decrease to a much smaller interval compared with the manual driving vehicles. In the initial stage, the platoon is designed for the same type of vehicles (e.g., heavy truck) within a single automobile manufacturer. That is because the same types of vehicles have similar mechanical characteristics and are able to accelerate or brake simultaneously. According to the principle of aerodynamics, the interval between the same vehicles would approximately remain constant, which would reduce the risk of a crash [2,3]. Subsequently, plenty of researches and projects have been carried out to address the more complex scenario of *dynamic* platoon formulation, in which vehicles are able to converge and depart throughout their journey. Generally speaking, there are three different types of AVP [4]:

(i) Scheduled AVP, SAVP: PL and PMs are predetermined before the journey starts. The number of vehicles remains the same throughout the entire trip. Variability is much less than the dynamic platoon mode. Thus, it is often referred to as *static AVP*.

(ii) Real-time AVP, RAVP: vehicles announce their trips when they are enroute. They can get in touch with the nearest preexisting scheduled AVP, request to join or leave at any time.

(iii) Opportunistic AVP, OAVP: vehicles in a close proximity to each other can dynamically form a platoon spontaneously. This kind of self-organized convoy is also denominated as on-the-fly or ad-hoc platoon.

RAVP is a much more preferred transportation pattern to the commuters since single-vehicle has the chance to enjoy the advantage of platooning. However, this brings new challenges either to the platoon management or to the information security. Firstly, SAVP members can be identified at the very beginning, and they remain constant until the journey is over, as there is no need to worry about the sensitive information leakage from an inner platoon member. However, there may be malicious vehicles who want to join into the RAVP, some of which may want to permeate into the platoon and launch some cyber-attacks (e.g., Sybil Attack, bogus information) to damage the interest of the other vehicles, some of which may act common but try to sniffer the privacy of the platoon even after they leave. Secondly, the large scale of vehicles with fast maneuverability also brings tremendous communication overhead to the centralized server, which makes it vulnerable to the Distributed Deny-of-Service (DDoS) attacks. Also, the blockchain can be leveraged to build a decentralized RAVP management system on the basis of 5G base station (A.K.A gNodeB, gNB). However, this would introduce a great deal of computation and storage overhead.

As a fundamental function, platoon management has been widely studied. Based on the existing works, platoon management can be categorized into protocol and strategy [5]. The protocol approach aims to tackle the challenge by proposing corresponding protocols related to different network layers [6–8], while the strategy approach includes maximizing platoon size and the platoon lifetime [9]. These works enrich the research in platoon management. However, none of the classical researches have considered the abovementioned security issues. Motivated by removing the obstacle of applying blockchain into RAVP management scenario, we propose a membership management scheme for 5G-enabled RAVP by integrating revocable attribute-based encryption (RABE) and blockchain, which is named as GAP-MM. Our main contributions can be summarized as follows:

(i) We propose a RAVP membership management scheme by integrating blockchain and revocable attribute-based encryption. Platoon leader can distribute a "*ticket*" to the identified individual vehicle to let it join the RAVP, or revoke the attribute after the individual vehicle leaves the RAVP, wherein the individual vehicle can join and leave the RAVP at any time.

(ii) A smart contract has been designed to realize vehicle join/leave automatically. To optimize the energy consumption either to the platoon leader or to the gNB, we also modify the block content and integrate the proposed GAP-MM into the Simulation of Urban MObility (SUMO) platform. The evaluation result indicates that the revised blockchain outperforms the other similar works in terms of gas or energy consumption.

(iii) We design a hybrid cryptography scheme to protect the privacy of RAVP. Besides identity key, individual vehicle would obtain another ABE key for the use of communicating in the RAVP. We proved the security of our key distribution algorithm. Moreover, we also discuss the most common cyber-attacks to analyze the robustness of GAP-MM.

The rest of the paper is organized as follows. In Section 2, we present the state-of-the-art platooning management methods in general. Section 3 gives the relevant preliminaries. In Section 4, the definitions of the system model and security model are given, followed by the detailed GAP-MM in Section 5. The analysis of GAP-MM in terms of security and performance can be found in Section 6. Finally, the conclusion is given in Section 7.

2. Related Works

2.1. Blockchain for RAVP. Wagner et al. propose a physical action verification scheme with blockchain [10]. They mainly focus on integrity verification when the roadside unit (RSU) is absent. When malicious vehicles try to join or leave the platoon, the protocol proceeds only when the vehicles can be sensed in a certain range. Ledbetter et al. first consider the incentive mechanism for the PL in dynamic and heterogeneous platoon [11]. They estimate the petrol consumption and try to relate it with the service pay. Calvo et al. propose a blockchain-based secure communication scheme for connected vehicles. They utilize the ring-signature to verify the identity of the join in vehicles, then the information can be shared among authenticated vehicles through a multi-party smart contract. Besides, they introduce the micro-transactions concept to deal with the low efficiency of consensus in bitcoin network, whereas they only provide with theoretical analysis but fail to give the experiment evaluation [12]. Zhang et al. present an onionchain-based

VANET framework to integrate the traceability of intermediate variables generated during the transactions [13]. For the purpose of encouraging vehicles participating in the building of an effective vehicular announcement network, Li et al. propose a privacy-preserving blockchain-based incentive announcement network for communication of smart vehicles. Thus, we design our consensus phases based on Byzantine fault tolerates algorithm to satisfy the requirements of efficiency in the scenario of VANETs [14]. Kang et al. proposed an optimized consensus management using reputation and contract theory to tackle the challenge of voting collusion. They used delegated proof-of-stake to realize consensus [15]. Cheng et al. integrate attribute-based encryption with blockchain to balance the tradeoff between the availability and the privacy preservation on the Internet of vehicles (IoVs) [16]. Ying et al. considered the cost efficiency of the AVP system. They design a hybrid chain model. In which the public chain provides certification records, and all platoon communication records will be stored on the privacy chain and will be uploaded to the public chain as platoon operation incident records [17].

2.2. Membership Management in Unmanned Aerial Vehicles.

Nowadays, the rapid development of 5G communication also facilitates the researches on Unmanned Aerial Vehicles (UAVs, also known as drones). UAVs can also form a "cluster" for the purpose of improving the resource usage efficiency and mitigating the loss of secrecy. Intuitively, these two entities have a lot in common with each other. Some crucial differences hinder the implementation of UAV cluster solution directly into the RAVP scenario, and vice versa. Feng et al. propose a blockchain-based privacy preserving data sharing scheme for 5G-enabled UAVs [18]. They consider outsourcing ABE in their approach, which could delegate some heavy computation task to the edge server. However, all the drones have to be registered at the trusted authority at the initial stage. Besides, they do not consider the membership variation as well as the key revocation issue. Bera et al. also consider the secure data delivery and collection issue in the Internet of Drones (IoD) environment [19]. They use elliptic curve cryptography (ECC) and one-way hash function to realize access control. Unfortunately, they only consider adding new drones into a flying zone situation but fail to consider the revoking of the key of a drone. Recently, Tan et al. put forward another blockchain-based key management for Flying Ad-Hoc Network (FANET). Besides reconstructing the structure of the block, they also realize the dynamic membership management without communicating with the gNB [20]. This work also brings us a lot of inspiration. However, their approach is established on a strong hypothesis that beyond the drone cluster there also exist some head drones which are in charge of managing the key blockchain. Apparently, in the RAVP scenario, we are not able to construct such a hierarchical topology. Therefore, we have to redesign the membership management for the RAVP.

3. Preliminaries and Definitions

3.1. Ethereum.

Ethereum is the product of a smart contract ported to the blockchain. Ethereum expands the scope of application of smart contracts, evolving the blockchain from a purely distributed repository to an open, compilable blockchain development project. Smart contracts are executed by participating nodes using an operating system known as Ethereum Virtual Machine (EVM). Ethereum carries a powerful Turing-complete development language. The ETH protocol is based on a bitcoin protocol, and the mining node verifies the new block so that new transactions are generated. Miners use a consensus algorithm for mining and can obtain mining fees paid by the transaction sender.

Ethereum uses The Ethereum Greedy Heaviest Observed Subtree (GHOST) protocol for consensus and miners' reward. When a miner builds a block, it sends it with its PoW through the network. Within the 14 seconds of the consensus, each node will receive numerous blocks. Some of them are supposed to be generated at the same time. Thus, it keeps the first in its main chain and considers the others as Uncles (equivalent definition to orphaned blocks in Bitcoin). It is the chain that contains the more of Uncles (called the heaviest chain) that will be kept as the main chain at the end of the consensus. Finally, the miner gets a part of the reward of the Uncles. GHOST also rewards the Uncles of the accepted blocks in order to strengthen the system.

3.2. Elliptic Curve Digital Signature Algorithm (ECDSA).

ECDSA is the migration of DSA on the elliptical curve. ECDSA has two processes for digital signature and signature verification. The elliptic curve parameter is $T = (p, a, b, G, n)$, and the elliptic curve is defined as $y^2 = (x^3 + ax + b)\mathrm{mod}\,p$, where p is a large prime number, F_p is a finite field, a, b are integers, G is the base point on $E(F_p s)$, n is a prime number that is the order of the base point G, the private key of PL is d, the public key $Q = dG$, k is the chosen random integer, e is the value of the hash operation of the message, and m, r are the remainders of x to n in the point (x, y) on the elliptic curve.

3.3. ECDSA Signature Generation.

A signs the message m. The steps are as follows:

$$\begin{aligned}
&A \text{ select arandom integer } k \text{ in the interval } [1, n-1]\\
&k = \text{Random Integer}\,(1, n-1)\\
&kG = \left(x_1, y_1\right)\\
&r = x_1 \bmod n\\
&e = \text{Has}\ (m)\\
&s = (er)^{-1}(k + d) \bmod n\\
&\text{signature} = (r, s).
\end{aligned} \tag{1}$$

3.4. ECDSA Signature Verification. After B receives the signature data (r, s) of A, to verify the signature of A on message m, the following steps are required:

$$
\begin{gathered}
\text{Verify } r, s \text{ is an integer in the interval } [1,\ n-1] \\
e = \mathrm{Hash}(m) \\
w = (er)s\mathrm{mod}n = (k+d)\mathrm{mod}n \\
wG - Q = (kG + dG) - dG = (x_1, y_1) \\
v = x_1\mathrm{mod}n.
\end{gathered}
\tag{2}
$$

If $v = r$, accept the signature, otherwise abort.

3.5. Access Structure. Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $A \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ is monotone if $\forall$ B, C: if $B \in A$ and $B \subseteq C$ then $C \in A$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) A of nonempty subsets of $\{P_1, P_2, \ldots, P_n\}$, i.e., $A \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \backslash \{\varnothing\}$. The sets in A are called the authorized sets, and the sets not in A are called the unauthorized sets.

3.6. Bilinear Maps. We present a few facts related to groups with efficiently computable bilinear maps. Let G_1 and G_2 be two multiplicative cyclic groups of prime order p. Let g be a generator of G_1 and e be a bilinear map, e: $G_1 \times G_1 \longrightarrow G_2$. The bilinear map e has the following properties:

(1) Bilinearity: for all u, $v \in G_1$ and a, $b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$

(2) Nondegeneracy: $e(g, g) \neq 1$

We say that G_1 is a bilinear group if the group operation in G_1 and the bilinear map e: $G_1 \times G_1 \longrightarrow G_2$ are both efficiently computable. Notice that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

3.7. Deterministic q-BDHE Assumption. Let g, G_N be a cyclic group of order p, p is a prime, G is a generator of G, e is a bilinear mapping, e: $G \times G = G_N$, randomly select the random $r \in Z_p$, calculate

$$
\mathrm{Q} = \left(g, g^s, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^q}, g^{\alpha^{q+2}}, \ldots, g^{\alpha^{2q}}\right). \tag{3}
$$

If there exist no poly-time algorithm that can distinguish $e(g, g)^{\alpha^{q+1}s}$ and the random elements in G_N in polynomial time, the q-BDHE assumption holds.

3.8. System and Attack Models

3.8.1. System Model. Our system consists of five entities as demonstrated in Figure 1. We briefly introduce the function and feature of each entity:

(1) Platoon Leader (PL): PL is the core of a platoon. It responds for creating the new platoon group, distributing tickets for the vehicles that want to join the platoon. Communicating with external facilities and other *PL*s. Releasing commands in the platoon. *PL* is assumed to be fully trusted.

(2) Original Platoon Member (OPM): OPM is the member vehicle in the original platoon. *OPM* receives commands from the *PL* and reports the supplementary road information to the *PL*. *OPM* is assumed to be honest.

(3) New Platoon Member (NPM): NPM can be of heterogeneous types of vehicles. After authentication, the individual vehicle turns into *NPM*. It receives commands from the *PL*. When an *NPM* wants to leave the platoon, it makes an announcement and pays the platoon service fee to the *PL*. NPM is assumed to be a untrusted and selfish vehicle which tries to escape from the payment and may propagate bogus information in the platoon.

(4) Certificate Authority (CA): CA is in charge of releasing public/private key pairs for each vehicle. All the vehicles should register themselves at the *CA* before they enter the system. *CA* does not have to be online during the entire platoon journey. The authentication work is delegated to the Ethereum. *CA* is assumed to be fully trusted.

(5) Road Side Units (RSUs): RSUs has two functions. First, they work as the communication base station. They receive messages from PL, individual vehicles, and forward them accordingly. Second, Some of the *RSUs* work as the miner node of the blockchain. When transaction happens (ticket generation, ABE key revocation, etc.), these nodes would make consensus via proof-of-work (PoW).

(6) Ethereum (ETH): there are two ETH subsystems operating in our proposed scheme, namely, the offline *ETH* and the online *ETH*. When the platoon is in ongoing, all the communication inside a platoon will be recorded on the offline *ETH* for future use. Platoon, *PL*, *OPM*, as well as *NPM* registration process will be recorded on the online *ETH*. Online *ETH* is also responsible for platoon service fee payment when a *NPM* wants to leave the platoon. Besides, when the trip is over, all the communication information which were recorded on the offline *ETH* will be uploaded to the cloud server, and the hash value of the record will be packed into the online *ETH* block.

3.8.2. Attack Model. In our proposed scheme, it is assumed that a newly joined vehicle cannot be fully trusted, or even malicious. It has the ability to launch various kinds of attacks such as information sniff, replay attack, trojan inject, transmission delay, and message tamper with negligible delay. The OPMs can receive tampered messages from the nodes inside a platoon. We are not aware of the rate of how many messages have been tampered. However, tampered messages can be detected. Besides, we only consider the protocol attacks other than physical attacks, in which the attackers could jam the communication channel or damage

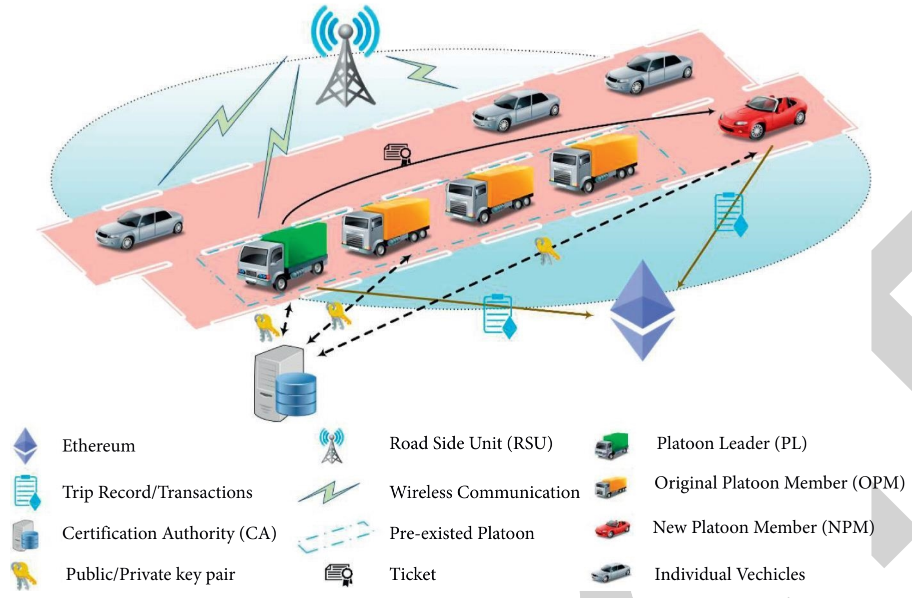


Figure 1: System model.

the communication infrastructure (e.g., Onboard Unit, OBU). These kinds of physical attacks can be provided by other specific techniques which are beyond our consideration.

An attacker could have multiple goals, such as sending wrong information in order to mislead platoon's decisions or the denial of the platoon system's services. Thus, it can conduct numerous attacks:

(1) Denial of service (DoS): DoS or Distributed DoS (DDoS) attack can be defined as consuming resources such as memory, the bandwidth of a server to prevent the normal users of obtaining the service. There are mostly two common ways of initialing a Dos/DDoS attack. First is to use SYN flood to attack a target server, and second is to take advantage of the protocol flaw. DoS/DDoS attacks are the most dangerous cyberattacks, The popularity of DoS attacks is because it is difficult to distinguish DoS flows from normal flows, thus the server has no time to activate defense mechanism. DoS attacks may also cause other computers on the same network of the target computer to be attacked. The bandwidth between the Internet and the local area network may be attacked and cause a lot of consumption, which not only affects the target computer but also affects other facilities in the local area network.

(2) Sybil attack: Sybil attack refers to the use of a small number of nodes in a social network to control multiple false identities, thereby using these identities to control or affect a large number of normal nodes of the network. Our proposed scenario is a peer-to-peer (P2P) network indeed. It is often the case that old vehicle nodes leave or new vehicle nodes join in the platoon. In order to maintain network stability, the same data usually needs to be backed up to multiple distributed nodes. This is the data redundancy mechanism. Sybil attack is an effective means of attacking data redundancy mechanisms. For example, if the traffic control center receives a message of a traffic accident, it will forward the message to the vehicles in the system to help them in the re-planning path. Whereas, the attack may send a piece of fake information to mislead the decisions given by the traffic center. To perform the Sybil attack, the attacker may use fake identity, or even stolen identity, which is not well detected when the attacker destroys or invalidates the original node.

(3) Spoofing attack: the difference between Sybil attack and spoofing attack is that the attacker tries to generate a number of fake identities in Sybil attack, while in the spoofing attack, the attacker tries impersonating the identity of an existing user to use his or her privilege.

(4) Message substitution attack: in this attack, the malicious user plays a man in the middle role, he or she eavesdrops on the open communication channel and tries to sniffer the information from the sender. After that, he or she modifies the original message and then sends it to the receiver. Then, the altered message will mislead the receiver.

(5) Replay attack: a replay attack, also known as a repeat attack or playback attack, is a form of network attack in which valid data transmission is maliciously or fraudulently repeated or delayed. It is mainly used for the identity authentication process and destroys the correctness of the authentication. Replay attacks can be performed by the initiator or by an enemy that intercepts and resends the data. The attacker uses network snooping or other means to steal authentication credentials and then resends it to the authentication server. For example, some systems simply encrypt the authentication information.

However, although the attacker cannot eavesdrop on the password, they can intercept the encrypted password and then replay it, so that this method can effectively attack. For another example, suppose that in the online deposit system, a message indicates that the user has taken a deposit, and the attacker can send the message multiple times and steal the deposit.

3.9. 5G-Enabled RAVP Membership Management Scheme

3.9.1. Design Goal. The main goal of our approach is to create secure virtual zones in real-time platoon environments. Each PM must communicate only with other PMs or PL of its zone, and consider every other vehicle as selfish and untrustworthy nodes. PL is able to communicate with the outside nodes (e.g., RSU, PLs from other platoons). We call these zones half-sealed bubble, where all its members can trust each other in the bubble. It is protected and inaccessible for nonmember vehicles. In order to achieve such a system, we rely on a public blockchain that implements smart contracts. We use a public blockchain other than a private chain in order to make the system open to any vehicle. It means that when a preexisting platoon is on the road, other single vehicles that want to join the platoon could find a way for communicating with the group, which would further improve planning the route of the vehicle and enhancing the practicality of the platoon.

Communications in the system are considered as transactions and must be validated by this private blockchain. For example, if vehicle A sends a message to vehicle B, then (1) A sends the message to the private blockchain, (2) if the blockchain authenticates A, it validates the transaction. Finally, (3) B can read the message. Communication among vehicles also needs to consume cryptocurrency. We separate the cryptocurrency used in private chain from public Ethereum. The cryptocurrency in the private chain can also be recognized as "*Gas*." However, it cannot be spent in a public chain. When a vehicle joins a platoon. An amount of private cryptocurrency will be allocated to it. Yet, the private cryptocurrency will automatically vanish when the vehicle leaves the platoon. Just like the private IPv4 addresses cannot be recognized on the Internet.

3.9.2. System Overview. We describe the system procedure of our proposed GAP-MM in this section. According to the timeline of the trip, the procedure can be categorized into before travel, during travel, and after travel:

(1) Before travel: some initialization work has to be done in this procedure. First, every vehicle has to register themselves on the Ethereum to get the corresponding public address. CA is the response for distributing public/private key pairs to each vehicle. Then, a vehicle is designated to be the leader of a platoon. The PL will generate *ticket* for other vehicles. Vehicles who got the *ticket* are able to join the team to formulate the original platoon.

(2) During travel: when the platoon is on the way, PL is able to communicate with PLs from other platoons and other facilities. NPM can only communicate inside the bubble. During the entire trip, communication information will be recorded in the PL through private Ethereum.

When a new vehicle wants to join in the platoon. It has to provide the identity to the PL. PL will check the applicant and release the *ticket* to the vehicle, then the new vehicle can join the platoon. All these steps will be recorded on the public Ethereum by using smart contract. Afterward, if the vehicle wants to leave the platoon, it has to communicate with the PL and finish the platoon service payment, which is also guaranteed by the smart contract.

(3) After travel: finally, when the platoon arrives at the destination address, PL will upload the entire travel records to the cloud, generate the corresponding hash to form a transaction, then upload it to the public Ethereum. If someone wants to check the traveling information, he or she will check if the record has been tampered.

3.9.3. Details of GAP-MM Scheme. In order to implement our protocol, we utilize the basic idea of constructing a "bubble" area for each platoon. The basic "bubble" restricts that communication in the interior area [21]. We made some modifications to let the PL communicate with external nodes. Besides, we also design the incentive mechanism. Our proposed GAP-MM protocol contains four modules:

(1) PL register: PL initials a new platoon creation procedure and adds itself into the platoon;

(2) Platoon ticket generation: PL generates tickets for the OPM as well as the NPM, and then OPM/NPM registers with an exclusive ticket into the platoon;

(3) Inter/Intra-platoon communication: authenticated OPM/NPM can communicate with each other inside the platoon, PL can also communicate with leaders of other platoons or gNBs;

(4) PM leaving: if an NPM wants to leave the platoon, it will make payment transactions with the PL.

Each Ethereum account has a pair of public and private keys, and the account node can use them to initiate transactions in the blockchain. PL sends a transaction which contains the *PlatoonId* and PL's identifier named *VehicleId*. The smart contract checks the uniqueness of them. As shown in Algorithm 1, if the smart contract determines that the vehicle address or PlatoonId has been registered, the registration agreement is terminated immediately. If it is a newly registered PL account and the *PlatoonId* has not been used, registration is allowed. After the PL is successfully registered, its private key can generate valid tickets. The PM provides the *VehicleId*, *PlatoonId*, and the blockchain account public address, then the PL integrates the data and digitally signs it with the private key. The signed data are returned to the PM as a ticket. The PM sends personal

VehicleId, *PlatoonId*, blockchain account public address and ticket to the smart contract. The smart contract checks the uniqueness of them and verifies the validity of the ticket. If the verification succeeds, the smart contract will complete the registration. This registration process is shown in Algorithm 1.

PL and PMs can perform intra-platoon communication. As shown in Algorithm 2, only the PL can perform inter-platoon communication. The smart contract automatically controls this restriction. Both PL and PM can apply for withdrawal after paying the required cost. As shown in Algorithm 3, according to the actual situation, we set the payment rules: if the driving distance of PM is less than 10 km, the PM needs to pay corresponding PL 1 Gas; if it is farther than 10 km, then according to the formula fee = [1 (distance − 10) ∗ 0.5]Gas calculates the required cost. Upon completion of the payment, the smart contract marks the member as being dequeued for later re-registration.

3.9.4. Attribute Generation and Revocation. Let G_1 be a bilinear group of prime order p, and let g be a generator of G_1. In addition, let e: $G_1 \times G_1 \longrightarrow G_2$ denote the bilinear map. A security parameter λ determines the size of the groups. For an *LSSS* structure $(\mathbb{M}, \rho)$, $\mathbb{M}$ is the $l \times n$ matrix, $\rho(i)$ is the attribute related to i-th row. Message is encrypted under $\mathcal{S}$. The vehicle can decrypt the ciphertext only if its attribute set $\mathcal{S}$ could satisfy the access structure $(\mathbb{M}, \rho)$. That is to say, there exists a coefficient $\omega_i \in Z_p$ satisfies $\sum_{\rho(i) \in \mathcal{S}} \omega_i \mathbb{M}_i = \overrightarrow{1}$.

Setup $(V_{id}, \mathcal{H}(x), a, b, y) \longrightarrow$ (PK, MK): define the universe of attributes $U = \{1, 2, \ldots, m\}$, and vehicle identity universe as $\mathcal{I} = \{1, 2, \ldots, n\}$. Now, for each attribute $i \in U$, choose a number t_i uniformly at random from Z_p. The identity of the vehicle is defined as $V_{id} \in \mathcal{I}$. Select a secure one-way hash function $\mathcal{H}(x)$ which could map the arbitrary length of string to an element on Z_p. Randomly select a, b, y in Z_p. The public parameters PK is formed as

$$\begin{aligned} &\langle T_1 = g^{t_1}, \ldots, T_{|U|} = g^{t_{|U|}}, g_{V_{id}} = g^{b\mathcal{H}(V_{id})}, \\ &Y = e(g, g)^y, (h = g^a, g_i = g^{a_i}, i \in U) \rangle. \end{aligned} \tag{4}$$

The master key MK is

$$t_1, \ldots, t_{|U|}, a, b, y. \tag{5}$$

Key Generation (MK, PK, V_{id}, $(\mathbb{M}, \rho)$) $\longrightarrow$ SK: this algorithm takes as input the access structure $(\mathbb{M}, \rho)$, PK, MK as well as the identity of the individual vehicle V_{id}. $\mathbb{M}$ is an $l \times n$ matrix, $\rho(i)$ denotes the attribute of the i-th row. Randomly choose vector $\overrightarrow{v} = (s, v_2, v_3, \ldots, v_n)$. Then we integrate the V_{id} into the user's private key. The private key is formed as

$$SK = \langle C_0 = g^{y + b\mathcal{H}(V_{id})}, C_1 = g^{b\mathcal{H}(V_{id})}, C_2 = V_{id}, K = g^{y/a + b\mathcal{H}(V_{id})}, K_i = g^{\overrightarrow{\mathbb{M}}_i \times \overrightarrow{v} ab\mathcal{H}(V_{id})/t_{\rho(i)}} \rangle. \tag{6}$$

Encryption (M, $\mathcal{S}$, PK, s, $\mathcal{R}$) $\longrightarrow$ CT: to encrypt a message $M \in G_2$ under a set of attributes $\mathcal{S}$, the encryption algorithm randomly chooses $s \in Z_p$, along with the revocation list $\mathcal{R}$ and the public key, then publish the ciphertext as

$$CT = \langle (\mathcal{S}, E_0 = g^s, E_0' = g^{as}, E' = MY^s, E'' = e(g, g_{n+1}^s), \{E_i = T_i^s\}_{i \in \mathcal{S}}, \mathcal{R}) \rangle. \tag{7}$$

Decryption (CT, PK, SK, $\mathcal{N}$) $\longrightarrow$ M: this algorithm takes as input the ciphertext CT, the vehicles private key SK, and the public key PK. Here $\mathcal{N} = U - \mathcal{R}$, which means the nonrevoked attribute sets of a vehicle. For ease of exposition, we present the simplest form of the decryption algorithm. The decryption performs as follows:

$$\begin{aligned} B &= \left(\prod_{\rho(i) \in \mathcal{S}} e\left(K_i, E_0 E_{\rho(i)}\right)^{\omega_i} \right) E'' \\ &= \left(\prod_{\rho(i) \in \mathcal{S}} e\left(g^{\overrightarrow{\mathbb{M}}_i \times \overrightarrow{v} ab\mathcal{H}(V_{id})/t_{\rho(i)}}, g^s g^{t_\rho(i)} \right)^{\omega_i} \right) e(g, g_{n+1})^s \\ &= e(g, g)^{\sum_{\rho(i) \in \mathcal{S}} \overrightarrow{\mathbb{M}}_i \times \overrightarrow{v} ab\mathcal{H}(V_{id}) \omega_i s} e(g, g_{n+1})^s \\ &= e(g, g)^{yabs\mathcal{H}(V_{id})} e(g, g_{n+1})^s. \end{aligned} \tag{8}$$

```
if Vehicle.exist InSmart Contract ( )ÚVehicle Address.exist In Smart Contract ( ) then
  return error
end
if Vehicle.type = PL then
    if PlatoonId.exist In Smart Contract ( ) then
      return error
    end
end
else
  if Vehicle.type = PM then
    if ! PlatoonId.exist In Smart Contract ( ) Úverify Ticket (ticket) = failed Ú ticket.used ( ) then
      return error
    end
  end
end
Registe Into Contract
```

Algorithm 1: The smart contract registration rules.

```
if sender.type = PM Ùsender.PlatoonId ! = receiver.PlatoonId then
    return error
end
sendMessage ( )
```

Algorithm 2: The smart contract communication rules.

```
if Vehicle.request Leave ( ) then
    if distance < 10 km then
      Vehicle.transferToPL (1Gas)
    end
    else
      Vehicle.transferToPL (1 + (distance − 10) ∗ 0.05 Gas)
    end
end
leave ( )
```

Algorithm 3: The smart contract leave and payment rules.

Then, we have

$$\begin{aligned} B' &= e\left(K, hC_1\right)e\left(\frac{C_0}{C_1, E_0}\right) \\ &= e\left(g^{y/a+b\mathscr{H}\left(V_{id}\right)}, g^{a}g^{b\mathscr{H}\left(V_{id}\right)}\right)e\left(g^{y}, g^{s}\right) \\ &= e\left(g, g\right)^{y}e\left(g, g\right)^{ys}. \end{aligned} \tag{9}$$

Finally, the message M can be recovered by

$$\begin{aligned} & E'\frac{B}{B'}\frac{e\left(\prod_{i\in\mathcal{N},i\neq V_{id}}g_{n+1-i+V_{id}}, E_0\right)}{e\left(g_{V_{id}}, \left(h\prod_{i\in\mathcal{N}}g_{n+1-i}\right)\right)} \\ &= Me\left(g, g\right)^{ys}e\left(g, g\right)^{abs\mathscr{H}\left(V_{id}\right)}e\left(g, g\right)^{as(n+1)}\frac{e\left(g, g_{n+1}\right)^{-s}}{e\left(g_{V_{id}}, h\right)^{s}} \\ &= M. \end{aligned} \tag{10}$$

Revocation: when a vehicle wants to leave the RAVP, it can send a request to the PL. After verifying the vehicle identity V_{id}, PL would renew the revocation list $\mathscr{R}$. Then, the corresponding attribute P_x will be revoked. If a malicious vehicle obtains the private key, since the identity of the vehicle has been embedded into it, every vehicle satisfies the following equation:

$$e(C_0, g) = e(C_1, g)e(g^y, g). \tag{11}$$

Therefore, we can get the vehicle id by using C_2 and then the initial revocation.

4. Security Analysis and Evaluations

4.1. Security Analysis. Our proposed scheme can handle different security challenges such as identification, Sybil attack, replay attack, and DDoS attack.

DoS/DDoS protection: a single platoon is a decentralized system of Ethereum which is born with immunity to DoS/DDoS attack. Outside nodes cannot communicate with the inside nodes. The authenticated vehicles could join the platoon and communicate with each other. Besides, communication in the platoon needs to consume gas, which increases crime costs.

Identification: each vehicle has an identity (*VehicleID* associated with a *PlatoonId* and to its public address (generated from its public key)). The signature in the *ticket* guarantees the trustworthiness of the identity. The *ticket* contains only 4 parts, namely, *PlatoonID*, *VehicleID*, the public address which is generated from its public key, and a signature of the concatenation of these three elements signed by the PL. There is no sensitive information involved. The *ticket* can be seen by everyone, whereas, no one could modify it since the adversary cannot obtain the private key of the PL. Each message is signed by the private key, which represents its identity. Thus, it can be easily identified.

Nonrepudiation: this characteristic is ensured by the signature. Each message is signed by the generator's own private key, and the private key is only known by its owner. As a consequence, the message generator cannot deny the fact of sending a message.

Scalability: the real-time platoon is built on the public Ethereum, which is a P2P network indeed. It has elastic scalability toward the server-centric network. In addition, the platoon has its own limits (up to 10 vehicles). The preexisted platoon has about 5 vehicles already. Thus, this scale of expansion is easy to implement in a distributed network structure.

Sybil attack protection: in our proposed scheme, every vehicle can only have one identity and one public/private key pair at a time. Every single message must be signed by the private key. Besides, in the initial phase, all vehicles have to register themselves to the public Ethereum, which prevents the attacker from creating fake identities.

Spoofing attack protection: same as the Sybil attack. The attacker cannot obtain the corresponding private key, so he or she cannot spoof other vehicle's identity. In addition, the Ethereum is able to track the history use of an identity. The abnormal use of identity can be distinguished.

Message substitution protection: the message substitution protection is realized by signature. The attacker cannot get the corresponding private key to sign the message. Besides, all the message will be packed into a block. The substitution operation will be identified through the hash check.

Message replay protection: all the messages can be regarded as transactions. Every transaction has its own timestamp. When the transaction needs to be packed into the block, the consensus phase is used to check the validity. Therefore, the attacker cannot initiate a replay attack. The replay message will be rejected by the consensus mechanism.

4.2. Complexity Analysis. To the best of our knowledge, there exists no similar work in applying blockchain to the management of real-time platooning. It is hard to compare our proposed GAP-MM with the related work directly. In this part, we present the complexity of our scheme, and describe the detail of fundamental cryptography computation cost.

The Datagram Transport Layer Security (DTLS) algorithm requires at least 5 messages to finish the handshake [21]. Besides, some other messages may be added as the Change Cipher Suite Message, and these association messages could possibly reach to 8 messages. Another approach of realizing authentication protocol for Wireless Communication Networks is built on Elliptic Curves Cryptography, which requires 5 messages for the initialization. In addition, a gateway is needed which would lead to the increase of messages. A robust authentication scheme presented by Jan et al. [22] can achieve the robust authentication for IoT scenario. It takes about 4 messages for the association phase.

The input/output is the most costly phase in the entire system. Reducing the number of messages could help decrease the consumption of the system, especially for resource constraint vehicles. Our algorithm only needs 2 messages to finish the negotiation. (1) Sending the transaction (message) from vehicle to blockchain and (2) Blockchain response to the transaction. Since the time of handshake decreases, the computation and energy cost decreases. In addition, we implement ECDSA algorithm to realize message authentication other than some pairing-based algorithms. The exponential computation cost is much cheaper and faster than the pairing computation. So the whole scheme is lightweight.

If we consider the authentication time of a message, it mainly depends on the block generate time of Ethereum (about 15 seconds). In order to further improve the efficiency, we utilize the hybrid blockchain idea. When the platoon is on the road, all communications will be recorded on the private chain. Each message will be regarded as a transaction and be packed up immediately. Since the intra-platoon is a relatively half-sealed zone, the communication among vehicles could not be an interference from the outside. Therefore, a private chain could be a proper way of recording the communication history of a platoon. The block generation rate is much faster than the public chain (less than 1 second), which is also equally superior to other authentication solutions.

4.3. Performance Analysis. AVP demands highly reliable real-time interactions. Therefore, we design a detailed experimental evaluation of each functional module; all of the experiments are the result of averaging 100 trials. The experiment environment consists of a Ubuntu 18.04 laptop equipped with an Intel Core i5-4590 CPU @ 3.30 GHz (4 virtual cores), 4 GB RAM, and a Ubuntu 18.04 workstation equipped with an Intel Core i5-7200U CPU @ 2.50 GHz (4 virtual cores), 8 GB RAM. The workstation is used to build the Ethereum simulation environment and run the smart contract, the laptop performs as the Ethereum node client.

The Ethereum environment is Ganache CLI, which is part of the Ethereum development tool Truffle suite, Ganache CLI uses ethereumJS to simulate full client behavior. It does not require the computational effort to mine the blocks, which makes it easier to test and use smart contracts written in the Solidity language. We developed the node client using C++ language. The interaction is realized by QJsonRpc, which is a Qt implementation of the JSON-RPC protocol (remote procedure call protocol). We can simulate the Ethereum operation locally by using these tools.

4.4. Financial Cost. Table 1 demonstrates the cost of platoon registration. Its unit is Wei, which is the smallest unit in the Ethereum token system (1 ETH = 10^5 Gas = 10^{18} Wei). We can see that registering a PM is more expensive than a PL. The reason is that the PM needs to upload the abovementioned ticket to the smart contract when registering. Smart contract stores the ticket backup to prevent duplicate registrations. The overhead of smart contracts grows with the amount of data increase. The cost of a smart contract is co-related with the data stored in it. An increase in the amount of data will cause the transaction initiator to pay more.

Table 1: Financial cost of registration.

Vehicle	PL	PM
Price (wei)	99551	122992

In Figure 2, the abscissa is the data size of the index message sent within the platoon or among platoons. Each pillar on the histogram indicates the cost of shipping the message. As the amount of messages increases, the cost per message sent increases. This is because we save the previously sent message data on the smart contract. When posting a new message, we connect the new message data with the previous message data through the method *String-Concatenate()* and then save it on the smart contract. This will lead to an increase in the total amount of this message data. As mentioned before, more data stored on the smart contract will lead to the greater cost of the transaction sender, so gradually connecting the message will lead to an increase in financial expenses. It can be inferred that when the PM withdraws from the platoon, the smart contract only clears the data in the platoon, which would not increase the amount of data, so there is no overhead in the transaction.

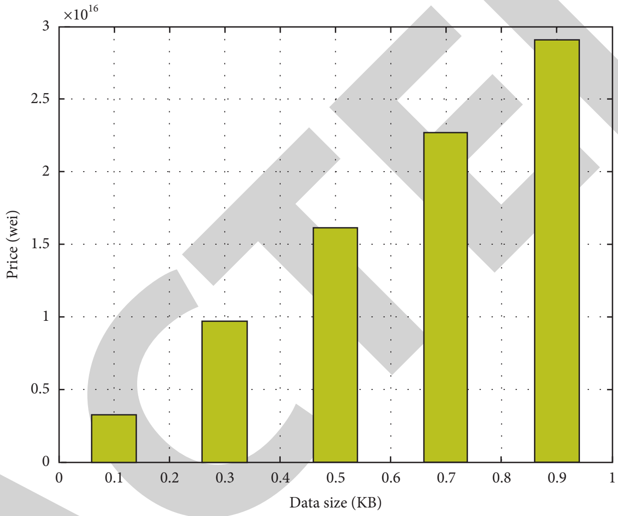


Figure 2: The cost of sending the Index message.

We construct the same data structure in the intra-platoon and inter-platoon messages, so the effect of sending messages within the platoon and among the platoon is the same for the data volume of the smart contract, and the overhead of sending the same amount of messages within and outside the platoon is the same. When the platoon leader and members send messages, the amount of data transmitted by the two is the same, and the code executed by the smart contract is the same.

5. Time Consumption

A recent study [10] measured the time overhead of its protocol. We compared the Join Platoon protocol simulated in Ref. [10] with the Platoon Registration protocol in our scheme. In Ref. [10], the time it takes to complete the Join Platoon protocol is linear to the number of blocks that need to be verified with a block size of 100 transactions. We simulate our Platoon Registration protocol in Ethereum by packaging and verifying blocks. As shown in Figure 3, both protocol speeds increase linearly with the number of blocks, and because the two use different verification algorithms, the structure of the blocks is different, and our protocol costs more time with the same number of blocks.

In the GANACHE CLI private blockchain simulation environment, newly created blocks can be added directly to the blockchain without mining verification. Each node needs to upload the message to the blockchain so that other users can download it for communication purposes. We designed an experiment where the independent variable is the length of the data to be uploaded, and a random function generates the content of the data. The unit of the data is KB. The dependent variable is the time, in seconds, required to upload data to the private chain. After 100 repeated experiments, we obtained the experimental results described in Figure 4. As the data length increases, the time required for data uploading increases linearly. The data need to be encoded and encrypted before going to the blockchain, which requires computational power. The node initiates the transaction to

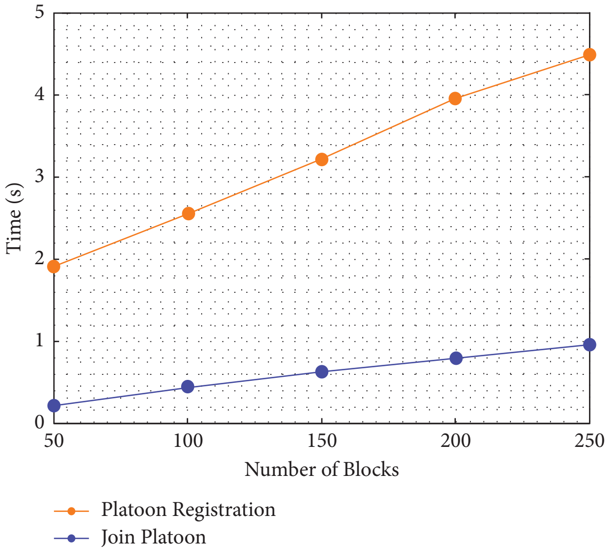


Figure 3: Speeds of two protocol.

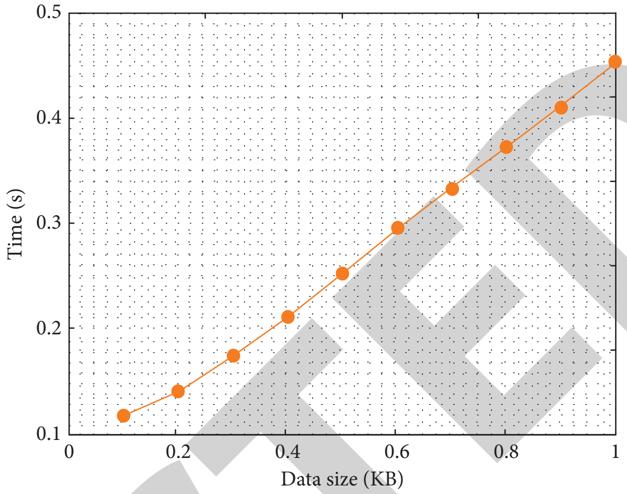


Figure 4: Data communication overhead.

upload the data, the smart contract decrypts the data after receiving the transaction and stores the original data on the smart contract to complete the transaction. However, smart contracts are very precious. It takes many resources to store data on smart contracts. The overhead of smart contracts increases with the amount of data that will be stored on the chain so that the time consumption will increase with the amount of data.

Table 2 demonstrates the average time required for PL and PM to dequeue and read the same amount of messages, respectively. PL and PM need the same time to leave a platoon and read common messages. In both cases, the amount of data transmitted by the two is the same, and the code executed by the smart contract is the same. There is no difference in the identity of the two when performing these two tasks.

We implement the revocable ABE algorithm in Python 3.7.3 with the charm library version 0.5.1. The SS512 asymmetric elliptic curve is used, in which the base field size is 512 bit, and the embedding degree is 2. All the experimental results are the mean of 100 trials. From Figures 5(a) to 5(d), we can find out that the time consumption grows linearly along with the attribute numbers. The most expensive calculation is encryption subroutine. When the PL wants to revoke the attributes given to an individual vehicle, it takes no more than 20 ms to do so, since the attributes that need to be revoked often contains platoon ID and the slot ID of the platoon only. Compared with the communication delay between the RSU and the vehicles, this computation overhead is acceptable.

6. DSRC Delay Experiment

To reach the efficiency requirements of the autonomous vehicles, we test the delay of DSRC wireless communication technology. In the process of sending each message, total delay D_{total} includes transmission delay D_{trans}, propagation delay D_{prop}, and reception delay D_{rece}, where the transmission and reception delays are equal. According to

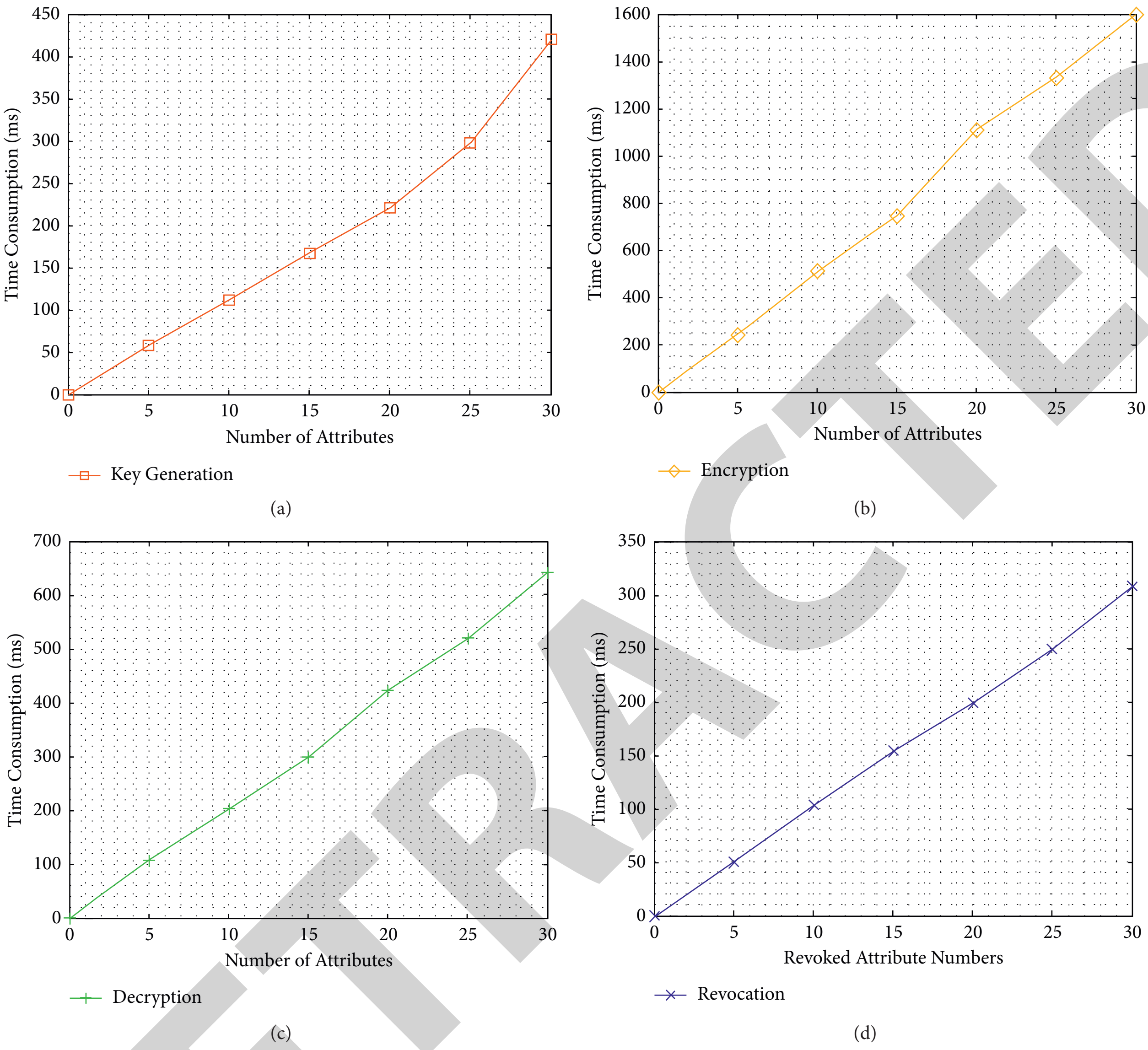
Table 2: Time cost of each function.

Function	Leave	Read message
Time (s)	0.437	0.052

$$\begin{aligned} D_{\text{trans}} &= D_{\text{rece}} = \frac{\text{frame size } (b)}{\text{transmission speed } (b/s)}, \\ D_{\text{prop}} &= \frac{\text{channel size } (m)}{\text{propagation speed } (m/s)}, \\ D_{\text{total}} &= D_{\text{trans}} + D_{\text{rece}} + D_{\text{prop}}. \end{aligned} \tag{12}$$

According to the 802.11p standard, we set the parameters into the formula for the situation when the network is more congested. In the formula, the Channel Size is considered to be the distance between PL and NPM, as shown in Figure 6. Assume that the DSRC devices of adjacent vehicles in a platoon is 5 meters apart. The theoretical calculation results are recorded in Table 3. As the distance between PL and NPM increases, the DSRC communication delay has a slight extension.

Take for example, a RAVP with 10 vehicles (4 vehicles are the OPMs, and the rest 6 vehicles are the NPMs). They all travel at the speed of 80 km/h. Then, the communication between the last vehicle and PL will not exceed 10 $*$ 100 = 1000 ms. That is to say, the whole platoon will travel about 22 m after 1000 ms, which is still in the coverage of an RSU. On the other hand, the transmission speed is set to be 8 Mbps and the propagation delay is set to be 2 μs within the range of 500 meters. The *ticket* size is 180 kb. According to the formulation of the above mentioned, we can figure out that the total delay is about 21.97 ms, which has an almost negligible effect on communication among vehicles.



(a)

(b)

(c)

(d)

Figure 5: RABE performance evaluation: (a) key generation, (b) encryption, (c) decryption, and (d) revocation.

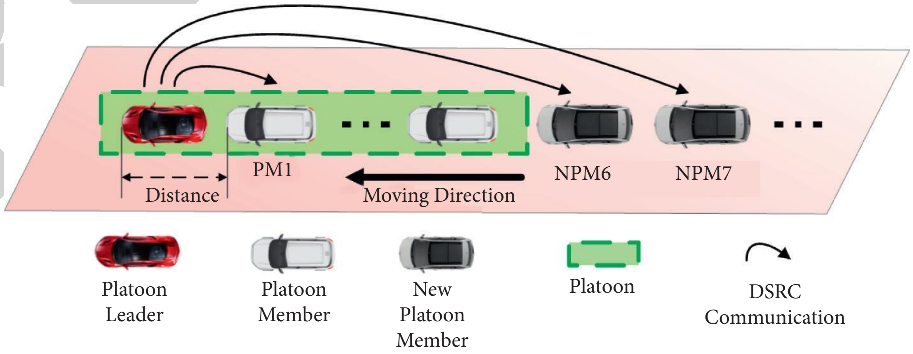


Figure 6: DSRC communication mode.

Table 3: DSRC delay between PL and NPM.

Index of NPM	6	7	8	9	10
Time (ms)	8.096083	8.096100	8.096117	8.096133	8.096150

7. Conclusion

An autonomous vehicle platoon management scheme is proposed based on Ethereum. A single vehicle can join and leave the platoon adaptively at any time. In addition, the transaction can be accomplished by using Ethereum. Meanwhile, the platoon leader's profit can also be guaranteed because of the blockchain feature. We analyze the security of the proposed scheme. The evaluation results indicate that our scheme is efficient. We list some quantitative index here. Firstly, considering the scenario of a RAVP with 10 vehicles traveling at the speed of 80 km/h, the total delay of generating the *ticket* is about 21.97 ms. Secondly, when an individual vehicle wants to leave the platoon, the revocation time is about 20 ms as there are two attributes that need to be revoked. Normally, the attributes that need to be revoked would be "platoon ID" and "slot ID." The block verification time of platoon registration and join platoon is about 2.5 s and 0.5 s, respectively, when the block number is 100.

Data Availability

The simulation experiment data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

A conference has previously been published [23].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Anhui Key Research and Development Plan under Grant 201904a05020091 and Key Scientific Research Projects of Chaohu University (Grant No. XLZ-201906).

References

[1] T. Reed, "Inrix Global Traffic Scorecard," 2019, https://static.poder360.com.br/2019/02/INRIX_2018_Global_Traffic_Scorecard_Report__final_.pdf.

[2] Scania, "The Scania Report 2018," *Annual and Sustainability Report*, 2018, https://www.scania.com/content/dam/scanianoe/global/pdfs/scania-annual-and-sustainability-report-2018.pdf.

[3] S. Tsugawa, "Final report on an automated truck platoon within energy its project," in *International Task Force on Vehicle Highway Automation 17th Annual Meeting*Sadayuki Tsugawa Meijo University Japan, Tokyo, Japan, 2013.

[4] A. K. Bhoopalam, N. Agatz, and R. Zuidwijk, "Planning of truck platoons: a literature review and directions for future research," *Transportation Research Part B: Methodological*, vol. 107, pp. 212–228, 2018.

[5] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 263–284, 2016.

[6] E. Moradi-Pari, H. Nourkhiz Mahjoub, H. N. Mahjoub, H. Kazemi, Y. P. Fallah, and A. Tahmasbi-Sarvestani, "Utilizing model-based communication and control for cooperative automated vehicle applications," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 1, pp. 38–51, 2017.

[7] P. Alberto, A. Pescapè, and S. Santini, "A collaborative approach for improving the security of vehicular scenarios: the case of platooning," *Computer Communications*, vol. 122, pp. 59–75, 2018.

[8] A. Sarker, H. Shen, M. Rahman et al., "A review of sensing and communication, human factors, and controller aspects for information-aware connected and automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 7–29, 2020.

[9] Y. Feng, D. He, and Y. Guan, "Composite platoon trajectory planning strategy for intersection throughput maximization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6305–6319, 2019.

[10] M. Wagner and B. McMillin, "Cyber-physical transactions: a method for securing vanets with blockchains," in *Proceedings of the 23rd IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2018)*, pp. 64–73, Taipei, Taiwan, December 2018.

[11] B. K. Ledbetter, S. Wehunt, M. A. Rahman, and M. H. M. Lips, "A protocol for leadership incentives for heterogeneous and dynamic platoons," in *Proceedings of the 43rd IEEE Annual Computer Software and Applications Conference (COMPSAC 2019)*, Milwaukee, WI, USA, July 2019.

[12] J. A. Leon Calvo and R. Mathar, "Secure blockchain-based communication scheme for connected vehicles," in *Proceedings of the European Conference on Networks and Communications (EuCNC 2018)*, pp. 347–351, Ljubljana, Slovenia, June 2018.

[13] Y. Zhang, J. Weng, J. S. Weng, M. Li, and W. Luo, "Onionchain: Towards Balancing Privacy and Traceability of Blockchain-Based Applications," 2019, https://arxiv.org/abs/1909.03367.

[14] L. Li, J. Liu, L. Cheng et al., "Creditcoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.

[15] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[16] L. Cheng, J. Liu, G. Xu et al., "SCTSC: a semicentralized traffic signal control mode with attribute-based blockchain in IoVs," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1373–1385, 2019.

[17] Z. Ying, M. Ma, and L. Yi, "BAVPM: practical autonomous vehicle platoon management supported by blockchain technique," in *Proceedings of the 2019 4th International Conference*

on Intelligent Transportation Engineering (ICITE), pp. 256–260, Singapore, September 2019.

[18] C. Feng, K. Yu, K. B. Ali et al., “Efficient and secure data sharing for 5G flying drones: a blockchain-enabled approach,” *IEEE Netw*, vol. 35, no. 1, pp. 130–137.

[19] B. Bera, S. Saha, A. K. Das, N. Kumar, P. Lorenz, and M. Alazab, “Blockchain-envisioned secure data delivery and collection scheme for 5G-based IoT-enabled internet of drones environment,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9097–9111, 2020.

[20] Y. Tan, J. Liu, and N. Kato, “Blockchain-based key management for heterogeneous flying ad-hoc network,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, 2020.

[21] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, “Bubbles of trust: a decentralized blockchain-based authentication system for IoT,” *Computers & Security*, vol. 78, pp. 126–142, 2018.

[22] M. Ahmad Jan, P. Nanda, X. He, Z. Tan, and P. L. Ren, “A robust authentication scheme for observing resources in the Internet of Things environment,” in *Proceedings of the 13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2014)*, pp. 205–211, Beijing, China, September 2014.

[23] W. Lobato, D. do Rosário, M. Gerla, and L. A. Villas, “Platoon-based driving protocol based on game theory for multimedia transmission over VANET,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM 2017)*, pp. 1–6, Singapore, December 2017.