BAVPM: Practical Autonomous Vehicle Platoon Management Supported by Blockchain Technique

Zuobin Ying [*] [†], Maode Ma[*], and Longyang Yi[†]

[*] School of Electrical & Electronic Engineering, Nanyang Technological University, 639798, Singapore

[†] School of Computer Science & Technology, Anhui University, 230601, Hefei, China

Abstract—**Autonomous Vehicle Platoon (AVP) is the most promising solution to various problems in the Intelligent Transportation System. However, how to effectively manage the join and leaving vehicles, and ensure the profit of the platoon leader remains an open problem. In this paper, we propose a dynamic AVP management protocol by implementing Ethereum. Vehicle who wants to join and leave the platoon has to communicate with the platoon leader, and all messages will be related to the corresponding transactions regulated by the smart contract. Considering the cost efficiency of the AVP system, a hybrid chain model is constructed. The public chain provides with certification records. All platoon message communication records will be stored on the privacy chain and will be upload to the public chain as platoon operation incident record. The evaluation result and security analysis indicate that our proposed scheme is practical for AVP scenario in terms of both efficient and secure.**

Keywords-autonomous vehicle platoon; platoon management; ethereum, smart contract

I. Introduction

Autonomous vehicle platoon (AVP) is expected to be the most promising solution towards traffic congestion, energy-saving, air pollution in intelligent Transportation System (ITS). Platooning is a driving pattern which allows vehicles to drive close together in a group. It decreases the space between the front and rear vehicles to a much smaller interval compared with the manual driving vehicles. Usually, there is a platoon leader (PL) in the group, the rest of the vehicles are recognized as platoon members (PMs). All the nodes in the platoon form a peer-to-peer (P2P) network. During the entire journey, the PL is responsible for collecting road information from the road- side unit (RSU) and other external vehicles and notifying PMs. PMs only have to follow the instructions given by the PL. All platoon members can communicate with each other either by using Dedicated Short Range Communications (DSRC) or through Cellular Vehicle-to-Everything (C-V2X). The development of semi-automated driving technologies, which can be collectively referred to as Cooperative Adaptive Cruise Control (CACC), as well as the popularization of autonomous vehicles (e.g., the Googles Waymo [1], GM's Cruise [2]), stimulates the progress of AVP industry. In the initial stage, the platoon is designed for the same type of vehicles (e.g. heavy truck) within a single automobile manufacturer. That's because the same type 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 [3], [4]. Subsequently, plenty of researches and projects have been carried out to address the more complex scenario of dynamic platoon formulation (e.g., *real-time platoon*), in which heterogeneous vehicles announce their trips when they are en-route. They can get in touch with the nearest platoon, request for convergence and departure throughout their journey.

Dynamic platoon is more preferred to the commuters since the single-vehicle has the chance to enjoy the advantage of the platoon. Whereas, this brings new challenges either to the platoon management aspect or to the information security aspect. Firstly, traditional AVP member can be identified in the very beginning, and remain constant until the journey is over, there is no need to worry about the sensitive information leakage from inner platoon member. Yet the member of the dynamic platoon can be from various kinds of vehicles, among which exist some malicious nodes, who might want to permeate into the platoon and launch some cyber-attacks (e.g., Sybil Attack, Distributed Deny of Service, bogus information) to damage the interests of other vehicles. Secondly, platoon leader may receive the requirements of merging into the platoon or leaving the platoon at any time. Apparently, it will cost the platoon leader additional resources to deal with the queries. Since the AVP is a resource constraint scenario, some incentive mechanisms should also be considered in addition to the fast authentication. Finally, some speculative vehicles may pretend to be honest when they want to join in the platoon, but denial the join-in fact when they want to leave the platoon so as to avoid paying the service fees. Situation maybe even complicated in the highway platoon when the communication facilities are not so much as in the urban area, opportunistic vehicles can also take advantage of the resource constraint feature to delay the payment on purpose.

As a fundamental function, platoon management has been widely studied. Based on the existing works, platoon management can be categorized into protocol and strategy [6]. The protocol approach aims to tackle the challenge by proposing corresponding protocols related to different network layers. The strategy way includes maximizing

978-1-7281-4553-2/19/$31.00 ©2019 IEEE

platoon size and the platoon lifetime. For example, Santini *et al* have proposed a distributed consensus algorithm to maintain stability in the platoon while maneuvers are performed and the topology of the platoon changes [7]. Heinovski *et al* have considered the platoon formation as an optimization problem and have proposed a centralized and a distributed approach using greedy heuristics to solve this problem [8]. Recently, some blockchain-based dynamic platoon management schemes have been considered. Wagner *et al* have put forward a physical actions verification scheme with blockchain [9]. 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* have first considered the incentive mechanism for the PL in dynamic and heterogeneous platoon [10]. They estimated petrol consumption and tried to relate it with the service pay. These works enrich the research in platoon management. However, none of the classical researches have considered the above-mentioned security issues. Besides, due to the large computation cost of distributed consensus algorithm, the two blockchain-based schemes also confess that the proposed schemes have low real-time performance, and suspect the reliability of implementing blockchain technology into large-scale platoon system. Motivated by solving the security problems, as well as removing the obstacle of applying blockchain into dynamic AVP management scenario. We propose a blockchain-based autonomous vehicle management scheme (BAVPM). Considering the cost-efficiency in the public blockchain is too much expensive, we introduce the hybrid dual blockchain mechanism. Authentications and data transmissions among PMs are recorded on the private blockchain and upload to the public blockchain after the journey. Our proposed BAVPM has two outstanding features in general:

- A semi-closed communication space for every single platoon is constructed. PMs can communicate with each other in the same platoon, only PL can communicate with facilities or vehicles from outside the platoon. This approach could significantly decrease the interference among different platoons, and in the meantime, effectively enhance the security within a platoon.
- The corresponding smart contract is designed to ensure the service payment could be accomplished even when the opportunistic vehicles try to disavow the platoon service fact. The evaluation results indicate that our proposed scheme is efficient.

The rest of this paper is organized as follows. In Section II, the definitions of the system model and security model are given, then the proposed BAVPM scheme is detailed in Section III. The complete analysis in terms of security and performance can be found in Section IV. Finally, the conclusion is given in Section V.

II. System Model and Backgrounds

A. System Model

Our system consists of five entities as demonstrated in Fig 1. We briefly introduce the function and feature of each entity.

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

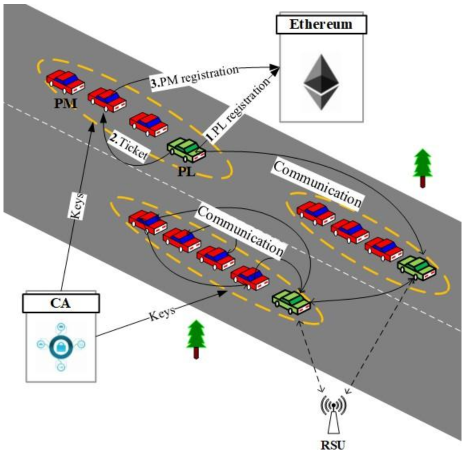

Figure 1. System model

Platoon Member: Platoon member (PM) can be of heterogeneous types of vehicles. After authentication, single-vehicle turns into PM. PM receives commands from the PL. When a PM wants to leave the platoon, it makes an announcement and pays the platoon service fee to the PL. PM is assumed to be dishonest. It tries to escape from the payment and may propagate bogus information in the platoon.

Certificate Authority: Certificate Authority (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.

Ethereum: Ethereum participants in the vehicle authentication, platoon management, and platoon service payment through the using of the smart contract. Ethereum is assumed to be fully trusted.

Roadside Unit: Roadside Unit (RSU) propagates traffic information. RSU does not have to participate in the platoon management procedure. It can be recognized just as a trusted communication facility.

B. 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. 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.

C. *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 define the elliptic curve as $y^2 = (x^3 + ax + b) mod\, p$, where p is a large prime number, F_p is a finite field, a, b are integers, G is the base point on $E(F_p)$, 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, and e is the value of the hash operation of the message m, r are the remainders of x to n in the point (x, y) on the elliptic curve.

1) ECDSA Signature Generation: A signs the message m. The steps are as follows:

A $select\ a\ random\ integer\ k\ in\ the\ interval\ [1,\ n-1]$.

$a = RandomInteger(1, n-1)$.

$kG = (x_1, y_1)$.

$r = x_1 \bmod n$.

$e = Hash(m)$.

$s = (er)^{-1}(k + d) \bmod n$.

$signature = (r, s)$.

2) 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:

Verify r, s is an integer in the interval $[1, n-1]$.

$\mathrm{e} = \mathrm{Hash(data)}$.

$w = (er)s \bmod n = (k + d) \bmod n$.

$wG - Q = (kG + dG) - dG = (x_1, y_1)$.

$v = x_1 \bmod n$.

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

III. BAVPM Protocol

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 [11]. We made some modifications to let the PL communicate with external nodes. Besides, we also design the incentive mechanism. Our proposed dynamic AVP management protocol contains four Modules:

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

2) Platoon ticket generation: PL generates tickets for the PM, and then PM registers with an exclusive ticket into platoon;

3) Inter-/Intra-platoon communication: Authenticated PMs can communicate with each other inside platoon, PL can also communicate with leaders of other platoons or RSUs;

4) PM leaving: If a PM 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 $PlattonId$ and PL's identifier named $ObjectId$. The smart contract checks the uniqueness of them. As shown in Algorithm 1, if the smart contract determines that the object address or $PlattonId$ has been registered, the registration agreement is terminated immediately. If it is a newly registered PL account and the $PlattonId$ has not been used, registration is allowed. After the PL is successfully registered, its private key can generate valid tickets. The PM provides the $ObjectId$, $PlattonId$, and the blockchain account public address, then the PL integrates the data and digitally signs it with the private key. The signed data is returned to the PM as a ticket. The PM sends personal $ObjectId$, $PlattonId$, 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, and the process of verifying the ticket is shown in Algorithm 5. 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.

Algorithm 1: The smart contract Registration rules

```
if $ObjectId.existInSmartContract() \lor$
$ObjectAddress.existInSmartContract()$ then
    return $error$
end
if $Object.type = PL$ then
    if $PlatoonId.existInSmartContract()$ then
        return $error$
    end
end
else
    if $Object.type = PM$ then
        if ! $PlatoonId.existInSmartContract() \lor$
        $verifyTicket(ticket) = failed \lor ticket.used()$
        then
            return $error$
        end
    end
end
$RegisteIntoContract()$
```

IV. Evaluation of BAVPM

A. Security Analysis

Our proposed scheme can handle with different security challenges such as identification, sybil attack, replay attack, and DDoS attack.

Algorithm 2: The smart contract Communication rules

```
if $sender.type = PM \wedge sender.PlatoonId\,!= receiver.PlatoonId$ then
    return $error$
end
$sendMessage()$
```

Algorithm 3: The smart contract Leave and Payment rules

```
if $object.requestLeave()$ then
    if $distance < 10\ km$ then
        $object.transferToPL\ (1\ Gas)$
    end
    else
        $object.transferToPL$
        $(1 + (distance - 10) * 0.05\ Gas)$
    end
end
$leave()$
```

Identification: Each vehicle will be issued an ObjectId as well as a *PlattonId* when it wants to join a platoon. Meanwhile, both of the two identities are registered on the Ethereum, which can be easily verified. Apparently, identification could be easily realized.

Sybil Attack: In our BAVPM, each vehicle is issued with only one identity and each vehicle can have only one private/public key pair. All the messages have to be signed by the private key associated with the vehicle. So the malicious user can not use fake identities.

Replay Attack: All the messages are considered as transactions. Every transaction has its own timestamp and requires for a consensus phase to prove the validation. Replay messages will be eliminated. Thus, replay attack can be prevented.

DDoS Attack: The blockchain feature has robustness against DDoS attack. Besides, our proposed BAVPM is a half- sealed area. PMs can only receive information from the PL. Moreover, since the transaction in the public chain is costly. If the attacker wants to initiate a DDoS attack, he has to consider the economic cost which will obviously reduce the DDoS attack intention.

B. Performance Evaluation

AVP demands high 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 trails. The experiment environment consists of a Ubuntu 18.04 laptop equipped with an Intel Core i5-4590 CPU @ 3.30GHz (4 virtual cores), 4 GB RAM and a Ubuntu

18.04 workstation equipped with an Intel Core i5-7200U CPU @ 2.50GHz (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.

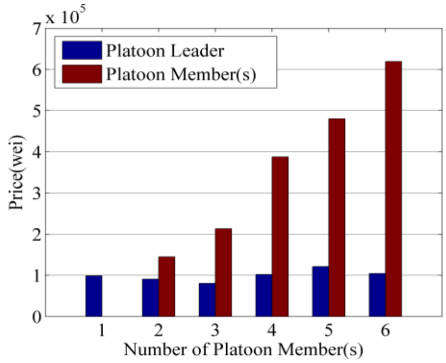

Figure 2. Fees for registration when the number of members is different

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.

1) Financial cost: Fig. 2 demonstrates the cost of platoon member registration. The abscissa is the number of members, and each platoon has only one leader. The ordinate is the total amount of registration for this platoon. Its unit is Wei, which is the smallest unit in the Ethereum token system (1 ETH = 105 Gas = 1018 Wei). Platoon leader registration fee is almost unchanged, yet the platoon members' registration fee increases linearly with the increase of the number. From the histogram of the number of the member(s) = 1, we can see that registering a PM is more expensive than a PL. The reason is that the PM needs to upload the above-mentioned ticket to the smart contract when registering. Smart contracts store 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 on it. An increase in the amount of data will cause the transaction initiator to pay more.

In Fig. 3, the abscissa is replaced by the index value of the message sent within the platoon or among platoons. Each point on the line indicates the cost of shipping the index 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 *StringConcatenate*() 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. 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 platoon is the same for the data volume of the smart contract, 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.

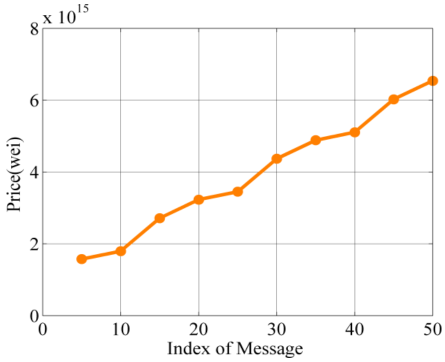

Figure 3. The cost of sending the Index message

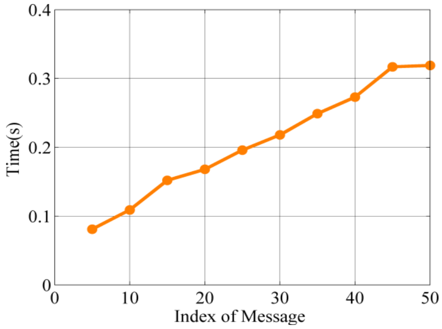

Figure 4. The time required to send the Index message

2) Time consumption: Table I demonstrates the average time required for PL and PM to register, dequeue, and read the same amount of messages, respectively. The PM requires a slightly higher time than the PL registration since it carries the ticket. The smart contract needs to verify the digital signature of the ticket using PL's public key. This verification process requires the calculation of data, resulting in increased overhead. Besides, PL and PM need the same time to leave the team and read the message. 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.

TABLE I. TIME COST OF EACH FUNCTION

Function	Register		Leave	Read message
	Master	Follower		
Time (s)	0.119	0.124	0.437	0.052

We pay more attention to the time overhead of sending messages. As shown in Fig. 4, as the amount of messages increases, the time it takes to send a message increases. When sending a message, it is necessary to connect the new message with the previously stored message and then store it in the s- mart contract. This connecting process is executed by the smart contract and consumes part of the time. The more messages stored in the past, the higher the difficulty of connecting, and the more computing power and time consumed.

V. CONCLUSION

A dynamic autonomous vehicle platoon management scheme is proposed based on Ethereum. A single vehicle can join and leave the platoon adaptively at any time. Additionally, 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.

ACKNOWLEDGEMENT

This work is supported by the NRF Systemic Risk and Resilience Planning Grant for the project of NRF2018-SR2001- 005 by National Research Foundation, Singapore.

REFERENCES

[1] Google, "Waymo". [Online]. Available: https://waymo.com/.

[2] GM, "Cruise Automation". [Online]. Available: https://getcruise.com. [3]Tsugawa S. Final report on an automated truck platoon within energy

[3] ITS project[C]. Presentation at the International Task Force on Vehicle- Highway Automations 17 th Annual Meeting. 2013.

[4] Scania Globle. "The Scania Report 2018,An- nual and Sustainability Report". Available: https://www.scania.com/content/dam/scanianoe/global/pdfs/scania - annual-and-sustainability-report-2018.pdf.

[5] Bhoopalam A K, Agatz N, Zuidwijk R. Planning of truck platoons: A literature review and directions for future research[J]. Transportation Research Part B: Methodological, 2018, 107: 212-228.

[6] Jia D, Lu K, Wang J, et al. A survey on platoon-based vehicular cyber- physical systems[J]. IEEE communications surveys & tutorials, 2015, 18(1): 263-284.

[7] Santini S, Salvi A, Valente A S, et al. Platooning maneuvers in vehicular networks: A distributed and consensus-based approach[J]. IEEE Transactions on Intelligent Vehicles, 2018, 4(1): 59-72.

[8] Heinovski J, Dressler F. Platoon Formation: Optimized Car to Platoon Assignment Strategies and Protocols[C]//2018 IEEE Vehicular Network- ing Conference (VNC). IEEE, 2018: 1-8.

[9] Wagner M, McMillin B. Cyber-Physical Transactions: A Method for Securing VANETs with Blockchains[C]//2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC). IEEE, 2018: 64-73.

[10] Ledbetter B, Wehunt S, Rahman M A, et al. LIPs: A Protocol for Lead- ership Incentives for Heterogeneous and Dynamic Platoons[C]//2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). IEEE, 2019, 1: 535-544.

[11] Hammi M T, Hammi B, Bellot P, et al. Bubbles of Trust: A decentral- ized blockchain-based authentication system for IoT[J]. Computers & Security, 2018, 78: 126-142.