Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8868656, 13 pages
https://doi.org/10.1155/2020/8868656

Research Article

BEHT: Blockchain-Based Efficient Highway Toll Paradigm for Opportunistic Autonomous Vehicle Platoon

Zuobin Ying[1,2] **Longyang Yi**[2] **and Maode Ma**[1]

[1]*School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore 639798*
[2]*School of Computer Science & Technology, Anhui University, China 230601*

Correspondence should be addressed to Zuobin Ying; james.ying@ntu.edu.sg

Received 25 March 2020; Revised 17 April 2020; Accepted 26 August 2020; Published 24 September 2020

Academic Editor: Ximeng Liu

Autonomous vehicle platoon is a promising paradigm towards traffic congestion problems in the intelligent transportation system. However, under certain circumstances, the advantage of the platoon cannot be fully developed. In this paper, we focus on the highway Electronic Toll Collection (ETC) charging problem. We try to let the opportunistic platoon pass the ETC as a whole. There are three main issues in this scenario. Firstly, the opportunistic platoon is temporarily composed; vehicles do not trust each other. Secondly, single vehicle may try to escape from the ETC charging by following the platoon. Finally, platoon members may collude with each other and try to underreport the number of vehicles in the platoon so as to evade payment. To solve these challenges, we propose a blockchain-based efficient highway toll paradigm for the opportunistic platoon. The driving history, credential information of every registered vehicle, is recorded and verified from the blockchain. A roadside unit (RSU) is adopted to distinguish the single vehicle from the platoon and in charge of lane allocation. Additionally, an aggregate signature is introduced to accelerate the authentication procedure in the RSU. We analyse the potential security threats in this scenario. The experimental result indicates that our scheme is efficient and practical.

1. Introduction

Vehicle platooning is one of the innovations in the automotive industry that is aimed at improving the safety, efficiency, mileage, and time of travel of vehicles while relieving traffic congestion, decreasing pollution, and reducing stress for passengers. Reliable vehicle platooning relies on the cooperation of multiple advanced technologies, including low-latency communication (e.g., 5G), proven autopilot system (e.g., level 5 full automation), multidriving model seamless switching (e.g., cooperative adaptive cruise control and self-govern autonomous driving), and, last but not least, flexible platoon management system [1]. Since autonomous vehicles are completely dominated by artificial intelligence, the credibility of the data will determine the safety of the entire platoon. However, data always suffer from various types of attacks, such as spoofing, data tampering, and compromised data integrity. Moreover, there also exist some network attacks, for example, Sybil attack and Distributed Deny of Service (DDoS). Therefore, how to guarantee the data as well as the network security in the autonomous vehicle platoon (AVP) becomes a considerable issue. In this paper, a blockchain-based approach is presented to deal with an interesting application issue in the *opportunistic platoon* scenario. We describe the issue and the relevant security and management problems to be solved first and then state our contributions.

The opportunistic platoon belongs to the dynamic platoon type [2]. There are two kinds of the dynamic platoon, namely, the *real-time platoon* and the *opportunistic platoon*. The difference between these two is that in the *real-time platoon* individual vehicles send a request to join a preexisting platoon, while there is no preexisting platoon in the *opportunistic platoon* scenario. Individual vehicles have to discover the vehicles with similar features (e.g., destination, vehicle type, and route) first and then try to formulate a platoon. Functionally speaking, they all accomplish the objective of platooning. However, from a security aspect, there was a huge difference between the two. Firstly, the *real-time platoon* is often launched by a company such as a supermarket or logistics company. The original vehicles can be regarded

Table 1: Security feature comparison.

	PL characteristic	PL variation	Original PM characteristic	New PM characteristic
Real-time platoon	Predesignated/fully trusted	Unchanged	Fully trusted	Semitrusted
Opportunistic platoon	Snap election/semitrusted	Changeable	—	Semitrusted

as fully trusted. Yet in the *opportunistic platoon*, vehicles do not even recognize each other and suffer from lack of trust. Secondly, compared with the *real-time platoon*, the *opportunistic platoon* is more changeable since vehicles may have different destinations, and the platoon leader has to manipulate more authentication tasks to realize vehicle joining or leaving. Finally, the platoon leader (PL) has to take on more work than the platoon member (PM). Moreover, according to aerodynamics, the platoon leader will sustain more air resistance. Thus, in the *opportunistic platoon*, no vehicle wants to be the leader. The security feature comparison is given in Table 1.

In this paper, we focus on a practical scenario in the highway. Nowadays, the highway tollgate still needs to decelerate before passing through the Electronic Toll Collection (ETC). After the ETC detects the vehicle, it raises the fence and releases the vehicle. The follow-up vehicle needs to be kept at a certain distance from the preceding vehicle in order to allow the ETC system to detect it. Thus, although the vehicle could enjoy the benefit from platooning, they would inevitably be separated and decelerate before passing the ETC. After that, vehicles have to reform a platoon. Obviously, this cumbersome operation will reduce people's enthusiasm for the platoon. Therefore, we considered the following question. *How to let an opportunistic platoon passing through the ETC as a whole?* For example, we can designate the PL to pay the ETC charge for the entire platoon. Then, other PMs could pass through the ETC without waiting or slowing down the speed. However, this is a complicated problem which includes not only theoretical problems but also the practical ones. We list the questions below:

(1) Since in the *opportunistic platoon* vehicles have no relationship with each other, if PMs try to escape from the part that they should pay, how to guarantee the rights of the PL?

(2) There exist both single vehicle and platoon in the highway. If a single vehicle tries to escape from paying the ETC charge by following the platoon, what should be done to prevent this situation?

(3) Generally speaking, the PL would not state that there are more vehicles than it actually exists in a platoon. Yet the PL may state less than the actual number of vehicles to cheat the ETC system. Since in our proposed scheme vehicles could keep a very high speed (e.g., 100 km/h) when passing the ETC, it is hard for the ETC to detect the accurate number of vehicles

Motivated by solving the abovementioned problems, we proposed the BEHT system: a blockchain-based efficient highway toll paradigm for opportunistic autonomous vehicle platoon. Our main contributions could be summarized as follows:

(1) The mutual mistrust issues in the *opportunistic platoon* are solved through blockchain. When the platoon passes through the ETC, the PL will pay for the entire platoon. Afterward, the PM could not repudiate to pay the part it should take. The smart contract will help to supervise the payment transfer from PM to PL

(2) A lane allocation mechanism is proposed to distinguish single vehicle with platoon so as to prevent single vehicle from escaping ETC charging by following the platoon

(3) We implement the aggregate signature into the *opportunistic platoon*. The PL needs to announce the number of vehicles in the platoon to the roadside unit (RSU) first, and all the members should sign on the announcement. Although the PL could underreport the number of vehicles, it is still possible to trace the actual number of vehicles in the platoon through the blockchain record

The rest of this 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; then, we give the details of the proposed blockchain-based efficient highway toll paradigm (BEHT) in Section 5. The analysis of BEHT in terms of security and performance can be found in Section 6. Finally, the conclusion is given in Section 7.

2. Related Works

The opportunistic platoon belongs to dynamic platoon management category. There are two different kinds of dynamic platoon. One is the *real-time platoon*, and the other one is the *opportunistic platoon* [3]. The opportunistic platoon refers to the vehicles that are close in a certain distance and have the similar interest or features with each other that form a temporary platoon without prior planning. The opportunistic platoon formation strategy is complicated. It requires not only the cooperation of vehicles in terms of maneuver but also the robust formation protocols. Besselink et al. give an overall review on cyber-physical control of road freight transportation [4]. They also discuss the possibility and precondition of forming an opportunistic platoon. Sokolov et al. considered the platoon formation maximization by coordinating the centralized routing and departure time. They also present a concrete simulation result as well as an

optimization model [5]. Since the deployment of platooning technology is not widespread, the potential benefits or opportunistic platoon has not been discovered. Therefore, some previous planning is required. Zeng et al. proposed a joint communication and control mechanism for wireless autonomous vehicular platoon systems; both the communication delay and the stability of control system are considered [6]. Through utilizing the Markovian jumping system theory, Wen and Guo proposed a sampled-data control system for connected vehicles subject to switching topologies, communication delays, and external disturbances [7]. Alam et al. discussed the significance of heavy-duty vehicle platooning that would help enhance safety and efficiency in global trade. They also evaluated the fuel saving, controller performance, and affectivity of changeable weather conditions. Moreover, the future of freight transportation system is given, in which the author believes that cooperation and platooning play an important role [8]. Boysen et al. investigated the platooning of trucks along an identical path since they found that the efficiency of platooning cannot only be guaranteed by the platooning technique but also be impacted by the platoon formation process [9]. Gong et al. developed a novel car-following control scheme for a platoon of connected and autonomous vehicles on a straight highway; they also constructed dual-based distributed algorithms to compute optimal solutions with proven convergence [10]. After that, Gong et al. proposed a series of research on how to optimize the AVP with human-driven vehicles in real world [11–13].

In the dynamic platoon management scenario, a newcomer may not willingly follow the protocols. For example, it may take advantage of the platoon to decrease the fuel consumption, but refuse to pay the platooning service charge, or it may propagate some phishing information to harm the platoon security. To deal with these security problems, some blockchain-based platoon management schemes have also been put forward. Wagner and McMillin proposed a physical action verification scheme with blockchain [14]. They mainly focus on integrity verification when the roadside unit 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. proposed a practical protocol for leadership incentives for a heterogeneous and dynamic platoon [15]. Through incentive mechanism, vehicles are encouraged to participate in the platoon leader election. Calvo and Mathar proposed a blockchain-based secure communication scheme for connected vehicles; they utilize the ring signature to verify the identity of the vehicles that joined, and then, the information can be shared among authenticated vehicles through a multiparty smart contract. But they only provided the theoretical analysis but failed to give the experiment evaluation. Moreover, they introduced the microtransaction concept to deal with the low efficiency of consensus in the bitcoin network [16]. Zhang et al. presented an onionchain-based VANET framework to integrate the traceability of intermediate variables generated during the transactions [17]. For the purpose of encouraging vehicles to participate in the building of an effective vehicular announcement network, Li et al. proposed a privacy-preserving blockchain-based incentive announcement network for communication of smart vehicles. They designed the consensus phases based on the Byzantine fault tolerance algorithm to meet the needs of reaching an agreement in a short period of time [18]. 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 [19]. Cheng et al. integrated attribute-based encryption with blockchain to balance the tradeoff between the availability and the privacy preservation on the Internet of Vehicles (IoVs) [20].

3. Preliminaries

3.1. Ethereum. Ethereum is an open-source, distributed computing platform based on a public blockchain with smart contracts' scripting capabilities [21]. With the use of smart contracts, Ethereum extends the range of application, making blockchains from purely distributed repositories to open, compilable blockchain development projects. Ethereum owns a powerful Turing complete development language, and it supports a modified version of the Nakamoto consensus through transaction-based state transitions. Miners use a consistent algorithm to mine and verify transactions for generating a new block. The Ethereum protocol moves far beyond currency. The currency named ether provides a liquidity layer to allow for efficient exchange of digital assets between public accounts and a mechanism for paying transaction fees.

3.2. Merkle Tree. A Merkle tree is a binary tree, in which every leaf node is labeled with the hash (e.g., SHA-256) of a data block, and every nonleaf node is labeled with the cryptographic hash by concatenating its child nodes as shown in Figure 1. The layer-by-layer operations from bottom to top, in turn, generate a unique node Merkle root. It is used to describe the integrity of all data block information stored.

Once the leaf node is modified, it will cause the change of the hash value on its parent node, which in turn affects the change of the Merkle root. According to its characteristic, in the blockchain, multiple transactions are used as data blocks of leaf nodes to build a Merkle tree. Any change in the transaction will cause a change in the Merkle root, and the integrity of all transactions can be verified by the Merkle root [22].

3.3. Elliptic Curve Digital Signature Algorithm. The Elliptic Curve Digital Signature Algorithm (ECDSA) offers a variant of the Digital Signature Algorithm (DSA) which uses elliptic curve cryptography. ECDSA has two processes for digital signature and signature verification [23]. The elliptic curve parameter is $T = (p, a, b, G, n)$, and the elliptic curve is defined as $y^2 = (x^3 + ax + b) \mod p$, where p is a large prime number, F_p is a finite field, a and 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 the PL is d, the public key $Q = G^d$, k is the chosen random integer, e is the value of the

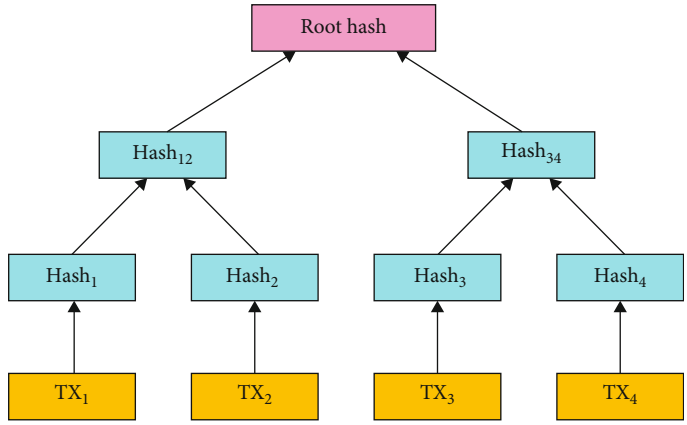

Figure 1: Merkle tree.

hash operation of the message m, and 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:

$$\begin{aligned} k &= RandomInteger[1, n-1] \\ G^k &= (x, y) \\ r &= x \bmod n \\ e &= Hash(m) \\ s &= k^{-1}(e + dr) \bmod n \\ signature &= (r, s) \end{aligned} \tag{1}$$

(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:

$$\begin{aligned} &Verify\ that\ r\ and\ s\ are\ integers\ in\ the\ interval\ [1, n-1] \\ &e = Hash(m) \\ &w = s^{-1} \bmod n \\ &u_1 = ew \bmod n\ and\ u_2 = rw \bmod n \\ &X = G^{u_1} Q^{u_2} \\ &If\ X\ is\ the\ point\ at\ infinity, then\ reject\ the\ signature. \\ &Otherwise, convert\ the\ x\ coordinate\ of\ X\ to\ an\ integer\ \bar{x}. \\ &v = \bar{x} \bmod n \\ &If\ v = r, accept\ the\ signature, otherwise\ abort. \end{aligned} \tag{2}$$

3.4. Aggregate Signature. The aggregate signature can create a signature on arbitrary distinct messages $M_i \in \{0, 1\}^*$ [24]. In this scheme, G_1 and G_2 are two multiplicative cyclic groups of prime order p. G_1 and G_2, their respective generators g_1 and g_2, the computable isomorphism Ψ from G_2 to G_1, and the bilinear map $e : G_1 \times G_2 \longrightarrow G_T$, with target group G_T, are system parameters. The scheme includes five algorithms: *KeyGen*, *Sign*, *Verify*, *Aggregate*, and *Aggregate Verify*.

(1) *Key Generation*. For a particular user, pick random $x \longleftarrow^R Z_p$, and compute $v \longleftarrow g_2^x$. The user's public key is $v \in G_2$. The user's secret key is $x \in Z_p$.

(2) *Signing*. For a particular user, given the secret key x and a message $M \in \{0, 1\}^*$, compute $h \longleftarrow H(M)$, where $h \in G_1$ and $\sigma \longleftarrow h^x$. The signature is $\sigma \in G_1$.

(3) *Verification*. Given the user's public key v, a message M, and a signature σ, compute $h \longleftarrow H(M)$; accept if $e(\sigma, g_2) = e(h, v)$ holds.

(4) *Aggregation*. For the aggregating subset of users $U \in$ Users, assign to each user an index i, ranging from 1 to $k = |U|$. Each user $u_i \in U$ provides a signature $\sigma_i \in G_1$ on a message $M_i \in \{0, 1\}^*$ of his choice. The messages M_i must all be distinct. Compute $\sigma \longleftarrow \prod_{i=1}^{k} = \sigma_i$. The aggregate signature is $\sigma \in G_1$.

(5) *Aggregate Verification*. We are given an aggregate signature $\sigma \in G_1$ for an aggregating subset of users U, indexed as before, and are given the original messages $M \in \{0, 1\}^*$ and public keys $v_i \in G_2$ for all users $u_i \in U$. To verify the aggregate signature σ

(i) ensure that the messages M_i are all distinct and reject otherwise

(ii) compute $h_i \longleftarrow H(M_i)$ for $1 \le i \le k = |U|$ and accept if $e(\sigma, g_2) = \prod_{i=1}^{k} e(h_i, v_i)$ holds

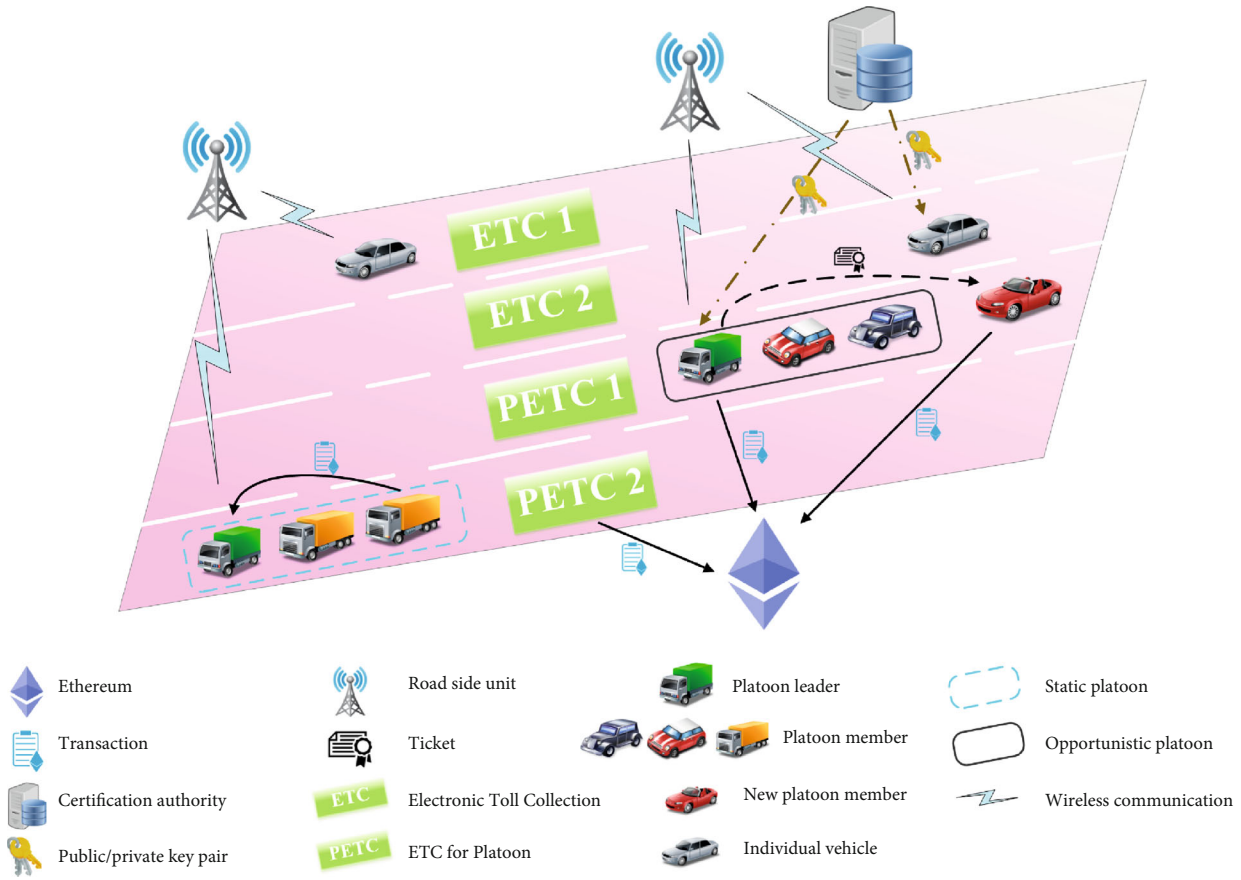

Figure 2: System model.

Using the properties of the bilinear map, the left-hand side of the verification equation expands:

$$e(\sigma, g_2) = e\left(\prod_{i=1}^{k} h_i^{x_i}, g_2\right) = \prod_{i=1}^{k} e(h_i, g_2)^{x_i} = \prod_{i=1}^{k} e\left(h_i, g_2^{x_i}\right) = \prod_{i=1}^{k} e(h_i, v_i), \tag{3}$$

which is the right-hand side, as required.

4. System Model

Our system consists of six entities: platoon leader (PL), platoon member (PM), roadside unit (RSU), certificate authority (CA), Ethereum, and Electronic Toll Collection (ETC) System, as shown in Figure 2.

(1) *Platoon Leader*. The platoon leader is the core of a platoon. It can create a new platoon, release commands in the platoon, communicate with RSU, and distribute tickets for the vehicles that want to join the platoon. However, it may release wrong commands on purpose.

(2) *Platoon Member*. New platoon member (NPM) can join into a platoon as a PM after authentication. The PM receives commands from the PL. The PM passes through the ETC behind the PL. The PM is assumed to be dishonest. It tries to escape from the payment in the platoon. In a static platoon, members are fixed. The opportunistic platoon allows any vehicle to join.

(3) *Roadside Unit*. The roadside unit releases traffic information and verifies the deduction transaction status. It can be recognized just as a trusted facility.

(4) *Certificate Authority*. Certificate authority is responsible for releasing public/private key pairs for each vehicle. All the vehicles should register at the CA before entering the blockchain. The CA does not have to be online during the entire platoon journey. The authentication work is entrusted to the Ethereum. The CA is assumed to be fully trusted.

(5) *Ethereum*. Ethereum is responsible for the vehicle authentication and ETC payment through the use of the smart contract. Its internal data structure Merkle tree can effectively verify transactions. The Ethereum is assumed to be fully trusted.

(6) *Electronic Toll Collection*. Electronic Toll Collection assigns the optimal ETC channel to platoons and deducts the registered owner's account without requiring them to park. It is a credible public facility like RSU. In particular, ETC for Platoon (PETC) is a channel specifically open for platoons.

5. Proposed BEHT Scheme

5.1. Design Goal. The goal of our proposal is to speed up the platoon's payment at the Electronic Toll Collection. In the real environment, the process of deducting the vehicle through the ETC, in turn, consumes a significant amount of time on the road. Our proposal puts several vehicles into a platoon with mutual distrust, passing through the ETC at just one time. In this way, the working time of the ETC is reduced exponentially, which dramatically improves its working efficiency.

To form a platoon not based on trust, we take advantage of the fact that the blockchain does not require trust, and the smart contract of the blockchain handles the process of building a platoon. Before the platoon travels to the ETC, the RSU can interact with the PL under our design protocol. The PL sends the result of the interaction to the PMs as instructions, which requires the function of intraplatoon communication in the platoon virtual environment. To prevent false messages issued by the PL, all communication contents will be permanently recorded on the blockchain. Then, the PL leads the PMs through the ETC in accordance under the requirements of RSU. The ETC charges the PL, and the required cost includes the sum of all vehicles in the entire platoon, which means that the PL pays for all the PMs at this time. At the end of the trip, the PM pays the service fee and ETC fee according to the transaction record to the PL through the blockchain.

In order to prevent PMs from refusing to admit that the PL paid for them, the signatures of all PMs will be aggregated to determine which vehicles have passed through the ETC. In addition, these payment transactions will also be recorded on the blockchain for later use as evidence.

5.2. Details of Scheme. In order to implement the scheme, we designed the platoon as a virtual area where private communication is possible. Moreover, we also proposed a protocol between RSU, platoon, and ETC to complete the payment through multiple interactions. Our scheme contains the following five modules.

(1) *PL Registration*. The PL applies for registering a platoon.

(2) *Ticket Generation*. The PL generates a dedicated ticket for NPM. The NPM uses the ticket to register into the platoon.

(3) *Communication*. The platoon members can communicate with each other while the PL can communicate with nearby RSUs through DSRC. All communication data will be backed up on the blockchain for later verification check.

(4) *Payment Interaction Protocol*. When the platoon is approaching the ETC, the RSU and the platoon perform multiple interaction confirmations according to the protocol we designed, and finally, the ETC completes the payment operation.

(5) *Credibility Mechanism*. This reputation mechanism is directly related to the amount of punishment in order to effectively reduce the probability of the platoon violating the protocol.

In the following paragraphs, we elaborate on these five modules in detail.

5.2.1. PL Registration. In our blockchain-based scheme, each vehicle needs to register in the blockchain to obtain account address and the public key and private keys from CA in advance. After successfully registering the blockchain account, the vehicle can initiate transactions and invoke smart contracts in the blockchain network.

Each device on the vehicle for ETC payment owns a unique identification code, which allows the vehicle to verify identity. The PL chooses a *platoonID* and provides its own ETC identification code *IC* to register in the smart contract that we deployed on the blockchain. If the smart contract detects that the *platoonID* and *IC* have never been used, the registration will succeed. The PL had applied for a virtual platoon on the smart contract.

5.2.2. Ticket Generation. The PL interacts with the vehicle who wants to join the platoon as a PM through the Dedicated Short-Range Communications (DSRC). After confirmation, it decides whether to allow the vehicle to join the platoon and generate a ticket to it. Vehicle should register in the blockchain network to obtain a blockchain account and public/private keys. The PM provides its own ETC identification code *IC*, *platoonID* of target platoon, and the blockchain account address *addr*. The corresponding PL of the platoon integrates the data and then digitally signs it by ECDSA digital signature algorithm E using the private key pk and returns the generated signature $sign = E_{pk}$ ($IC||platoonID||addr$) to the PM as a ticket. With the ticket, the PM is eligible to join this platoon.

The PM sends the ETC identification code *IC*, *platoonID*, blockchain account address *addr*, and ticket to the smart contract on the blockchain to register into the platoon. If the smart contract detects that the *IC* has not been registered, the *platoonID* exists, and the ticket has never been used, smart contract uses the *platoonID* to query the public key *PK* of the PL to verify the digital signature of the ticket by ECDSA digital signature verification algorithm V_{PK} ($IC||platoonID||addr, ticket$). If the verification succeeds, the PM is permitted to enter the platoon. All registration rules are shown in Algorithm 1.

After the PMs join into the platoon, the platoon is formed. In this scheme, the platoon includes static platoon and opportunistic platoon. All vehicles in the static platoon are unchanged and do not receive other vehicles to apply for entering the platoon. At this time, the PL does not distribute the exclusive ticket to the vehicle. The opportunistic

```
if IC.ExistInSmartContract()⋁(8)
      addr.ExistInSmartContract() then
          return Error(9)
      end.
if Vehicle.type = PL then
        if platoonID.ExistInSmartContract() then
          return Error(10)
        end
      end
      else
        if Vehilce.type = PM then
          if !platoonID.ExistInSmartContract()⋁
            VerifyTicket(ticket) = failed⋁ticket.Used() then
          return Error
          end
        end
      end
      RegisteIntoContract()
```

Algorithm 1: The Smart Contract Registration Rules

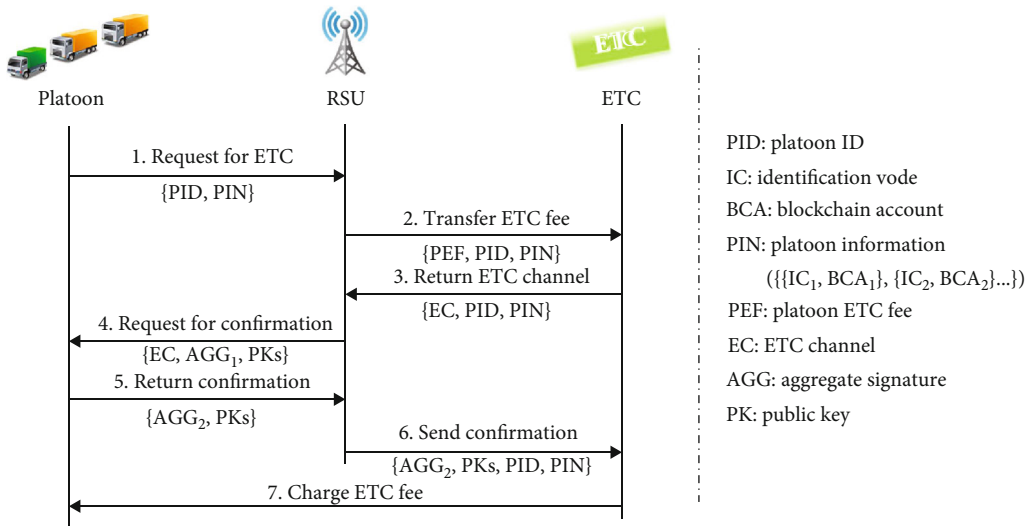

Figure 3: Interaction payment protocol.

platoon is a temporary set of multiple vehicles on the road, with substantial variability. Vehicles who want to join the platoon apply for the PL, and the PL returns ticket to each vehicle to join the platoon.

5.2.3. Communication. In our previous research results [25], the efficiency of uploading data to the blockchain is too slow, so we take a new approach that platoon members use DSRC to communicate with low latency directly and slowly back up the communication data on the blockchain permanently for later verification as a deposit. To achieve resource sharing, different platoons can also send and receive data through PLs using DSRC. Moreover, the PL can also communicate with nearby RSUs which send messages back, which enables multiparty data forwarding sharing.

5.2.4. Interaction Payment Protocol. In our scheme, when the platoon travels to ETC, in order to reduce the vehicle deceleration time significantly, the PL leads all PMs to pass ETC quickly and complete the deduction process at the same time. In such a short period, ETC cannot perform autosensing recognition for each vehicle. So the solution we adopted is that ETC only needed to identify the PL and deduct the costs of all platoon members from the PL. That is, the PL pays the ETC fee for PMs. The RSU, ETC, and platoon interact to complete the deduction protocol. Figure 3 demonstrates these interaction payment protocols. The specific processes are as follows:

(1) When each vehicle gets onto the highway and passes through the ETC, ETC identifies the vehicle's information and records the original station on the blockchain. On the highway, multiple vehicles build their platoons. When the platoon reaches about 1 kilometer before the ETC, the PL of the platoon sends a

request to nearby RSUs, including detailed vehicle information of the platoon

(2) Having obtained the platoon's information, the RSU queries the blockchain for the original station of each vehicle and calculates the cost that each vehicle should pay to the ETC. Combine the results into a set of data for each vehicle, such as

{plate number:A.0001, *IC*: 001122, *original station: New York station, terminal station: Washington station, ETC fee*: 10 *dollars}*

The RSU sends the total fee and information of all platoon members to the ETC system

(3) The ETC system would store the fee amount and return the assigned optimal ETC channel for the platoon to RSU

(4) The RSU adds the ETC channel into the data set and digitally signs each data with its private key for aggregating all the digital signatures together to generate an aggregate signature. The RSU sends the aggregate signature, the original data set for verification, and its public key to the PL as requesting for confirmation, which in turn forwards it to all PMs

(5) The PM verifies the aggregate signature and confirms the amount of the deduction. If the PM confirms that the amount of its deduction is correct, it will digitally sign the set of data with its private key and then send it to the PL. After the PL collects the digital signatures of all the PMs, it aggregates them to generate an aggregate signature. Finally, the PL sends the aggregate signature to the RSU as confirmation. The RSU sets a waiting time. If no confirmation response is received after the timeout, the RSU will retransmit, which is called the timeout retransmission mechanism

(6) After receiving the aggregate signature, the RSU queries the public key of each vehicle on the blockchain and combines the original data to verify the validity of the aggregate signature. If the verification passes, the confirmation step for the payment is completed. All confirmation records would be sent to ETC

(7) ETC would store all records in the blockchain and monitor the platoon for charging

The PL must lead the platoon to the designated ETC channel according to the instructions from RSU. The PL is driving in front of the platoon. The ETC automatically recognizes the ETC device of the PL then charges the PL according to the total cost fee sent from RSU before and upload the transaction record to the blockchain. In a short time, the platoon passed this ETC.

Based on the speed of the platoon, RSU estimates that the platoon has passed ETC to complete the deduction in a few minutes. At this time, the RSU queries the blockchain for a payment record and verifies the correctness of it. If the valid deduction information did not exist, the RSU communicates with the PL again to query the current platoon status and recomplete the payment protocol. At the end of the trip, the PM pays the service fee and ETC fee according to the transaction record to the PL through the blockchain. This incentive mechanism will promote more vehicles to undertake the tasks of the PL.

5.2.5. Credibility Mechanism. To maintain the orderly operations of the scheme, we will introduce the concept of vehicle credibility value (*CV*) and impose late penalties on vehicles that violate the protocol. The credibility value directly affects the penalty amount.

If the platoon does not interact with the RSU according to the protocol or fails to pass the designated channel based on the RSU's instructions, this will reduce the credibility value of each platoon member and impose a penalty charge according to the penalty standard. If the platoon does not provide an aggregate signature to ETC or RSU, the ETC cannot confirm each member of the platoon. That is to say, ETC only deducts the required fee of the PL, and PMs evade the deduction operation. According to the current highway penalty mechanism, the PM for this situation is charged on the farthest distance of the current highway, and the amount to be paid is recorded for later punishment. In our proposal, the credibility value of each platoon member will be reduced, and the penalty amount for breaching the protocol will be deducted. Simultaneously, if the platoon completes the protocol, the credibility value of the platoon members will also increase. The relationship between the credibility value and the penalty amount is shown in Figure 4. We set two thresholds for the penalty amount and the rapid penalty amount. The maximum credibility value is 100, and the minimum is 0. The specific segmentation function is as follows:

(1) $CV > 90$. The vehicle follows the protocol well and does not require additional deductions. It is the last piece with the green line in Figure 4.

(2) $50 \leq CV \leq 90$. The possibility of occasional mistakes in the platoon leads to a violation of the protocol. The amount of the deduction is linear, and its slope is small. It is the middle piece with the blue line in Figure 4.

(3) $CV < 50$. We can conclude that this vehicle often makes mistakes, even deliberately violates the protocol. We will make severe punishment for this kind of vehicle, and the penalty amount will increase exponentially with the decrease of *CV*. It is the first piece with the yellow line in Figure 4.

6. Simulations and Performance Analysis

6.1. Security Analysis. The scheme proposed in this paper enables the opportunistic autonomous vehicle to form a platoon and pass the ETC together. In this environment of mutual distrust, we focus on the attacks of platoon by bogus information injection and repudiation.

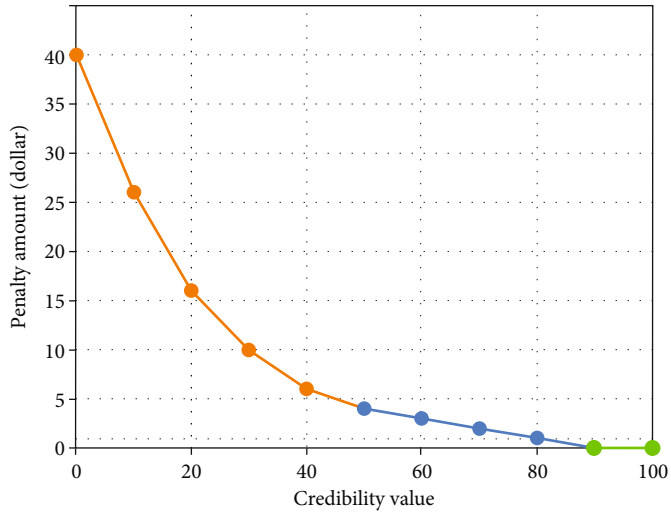

Figure 4: Penalty rules. Low credibility value leads to increased penalties.

6.1.1. Bogus Information Injection. Bogus information injections occur when the vehicle or RSU sends the wrong message. When a vehicle wants to join a platoon, it is possible that it just sent a false message to the PL remotely and is not nearby. In the ticket generation phase, the PL must interact with the vehicle to allow it to join the platoon. Therefore, the request of the attacking vehicle to join the platoon will be rejected if it is not in the sensing range.

In the communication of the scheme, bogus information injection will happen with many chances. The PL may send a fake message or instruction inside the platoon. After receiving the commands from the PL, the PM will use its sensing equipment to verify the feasibility of the commands and then decide whether to perform operations. At the same time, all messages inside the platoon will be recorded in the blockchain to prevent members from posting malicious messages.

In the interaction payment protocol, the data to be confirmed would be sent and received between the RSU and the PM, but it needs to be sent to the PL first and then forwarded to the PM by the PL. In our protocol, RSU digitally signs the data with its private key and generates an aggregated signature, ensuring that the PL cannot modify or falsify the data during this period. If the PL intentionally loses the digital signature, this will be detected by the RSU, and then, its credibility value will be lowered, and a penalty will be imposed on it. The RSU calculates the ETC fee for the vehicle. The vehicle's original station and terminal station on the highway are recorded on the blockchain. RSU deliberately adjusts the ETC fee privately and will be detected through traceability of the blockchain.

6.1.2. Repudiation. In the opportunistic autonomous vehicle scenario, repudiation occurs between mutually distrusted vehicles. They refused to admit having done something. Our scheme is based on blockchain, and repudiation is no longer a problem. Because the blockchain is immutable, any operation on the blockchain will be recorded for verification.

After the platoon passed the ETC, the PL temporarily paid the ETC fees for all PMs, and any PM may refuse to admit that it owed the PL some expenses. The traceability of the blockchain keeps the transaction of the PL payment permanently on the blockchain. The PL can download the transaction for use as a credential.

In the interaction payment protocol, when the platoon is passing ETC, ETC only needs to detect the PL and ignore the PMs to complete the payment transaction. However, at this time, there will be a vehicle outside the platoon closely following the platoon through the ETC to evade payment. ETC did not detect the vehicle, and the transaction for the vehicle deduction did not occur. After a few minutes, the RSU could not detect the transaction of the vehicle deduction on the blockchain. When the RSU communicates with the vehicle and requests to recomplete the protocol, the vehicle has passed the ETC, and the protocol cannot be completed. In this case, the vehicle violates the normal execution of the protocol. We will reduce its credibility value and make a fine. Moreover, for the penalty for the current highway penalty mechanism, the PM for this situation is charged on the farthest distance of the current highway for later punishment.

6.2. Experimental Evaluation. 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 an Ubuntu 18.04 laptop equipped with an Intel Core i5-4590 CPU @ 3.30 GHz (4 virtual cores), 4 GB RAM, and an 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 simulate the Ethereum environment and run the smart contract. The laptop performs as the Ethereum node client.

We use Ganache CLI to simulate the Ethereum environment, which applies ethereumjs to fulfill all Ethereum client behaviors. It does not require computational effort to mine the blocks, which makes it easier to run smart contracts written in the Solidity language and node clients using C++. The interaction between blockchain and node is realized by QJsonRpc, which is a Qt implementation of the JSON-RPC

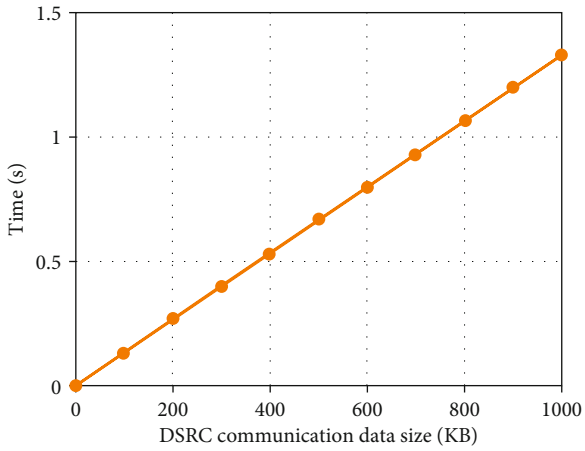

Figure 5: Time consumption of DSRC.

protocol (remote procedure call protocol). We can complete all the Ethereum operations using these tools.

6.2.1. Time Consumption for DSRC. In our scheme, DSRC technology is used in many scenarios, including data communication between the platoon members and RSUs. To reach the efficiency requirements of autonomous vehicles, we test the time consumption of DSRC wireless communication technology. According to the 802.11*p* standard, we set the parameters as ChannelDataRate = 6 Mbps and PropagationDelay = 2 μs in transmission range about 500 m. The theoretical calculation results are recorded in Figure 5. As the data size increases, the DSRC communication delay has a linear extension.

In the interaction payment protocol, the platoon initiates a request for ETC payment to RSU that would confirm the payment between them through DSRC data transferring. We calculate the data time consumption of each function, as demonstrated in Table 2.

6.2.2. Time Consumption for Blockchain. Figure 6 gives the time cost of platoon member registration. The abscissa is the number of members, including one leader. The ordinate is the time of registrations for this platoon. The time consumption of the PL and PM is almost constant. Therefore, the platoon members' registration time increases linearly with the number of platoon members. The histogram of the number of the member(s) = 2 demonstrates that registering a PM requires more time than a PL. That is because the PM should upload the ticket mentioned above to the smart contract when registering. The smart contract requires additional time to verify and store the ticket backup to prevent duplicate registrations.

All communication data in a platoon even through DSRC would be uploaded to the virtual platoon zone in the smart contract for later verification as a deposit. We designed an experiment where the independent variable is the length of data generated randomly. Its unit is KB. The dependent variable is the time required to upload and download data from blockchain's smart contract. After 100 repeated experiments, we obtained the experimental results described in Figure 7. As the data length increases, the time required increases almost linearly. The client should encode the data before calling function to the smart contract, which requires computational power. The smart contract decodes the data after receiving, and stores it on the smart contract. However, the overhead of the smart contract is considerable. It takes resources to store data on smart contracts, and the overhead increases with the amount of data. Therefore, the time consumption will increase with the data size. In the meantime, the time for downloading data is almost unchanged. Clients just initiate *eth_call* requests to the blockchain that would send corresponding data back, which consumes some query and network time. With the data size increasing, the time for downloading would increase little.

During the interaction payment protocol phase, the PL on behalf of the platoon agreed to deduct fees from RSU through ETC, and then, the platoon could pass ETC. After ETC deducts the platoon, it will upload transaction records to the blockchain for deposition inquiry as required. Ethereum's public chain has low efficiency and instability when linking new blocks. In this case, the platoon can query transaction records after a while. Figure 8 records the time for linking ten consecutive randomly selected blocks. The abscissa is each block index, and the ordinate is the consensus time consumed when the block was generated.

Table 2: Time consumption for each function in interaction payment protocol.

Feature	Request for ETC payment	Request for confirmation	Return confirmation
Data size (KB)	297	110	333
Time (ms)	0.396	0.147	0.444

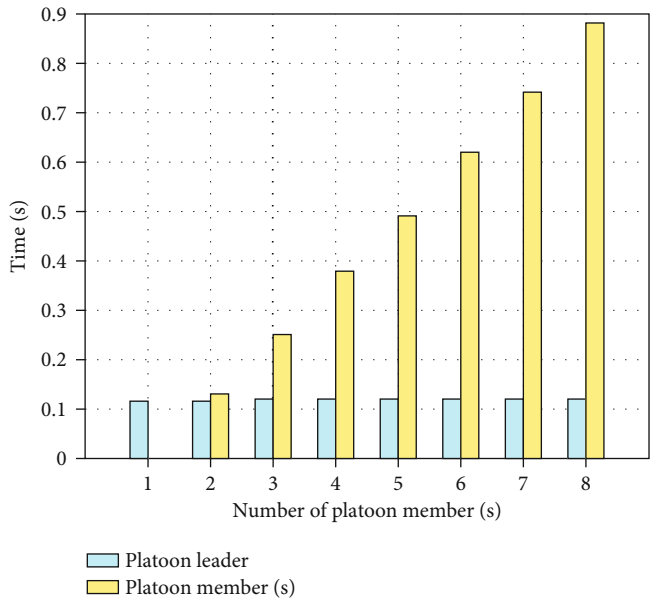

Figure 6: Registration time.

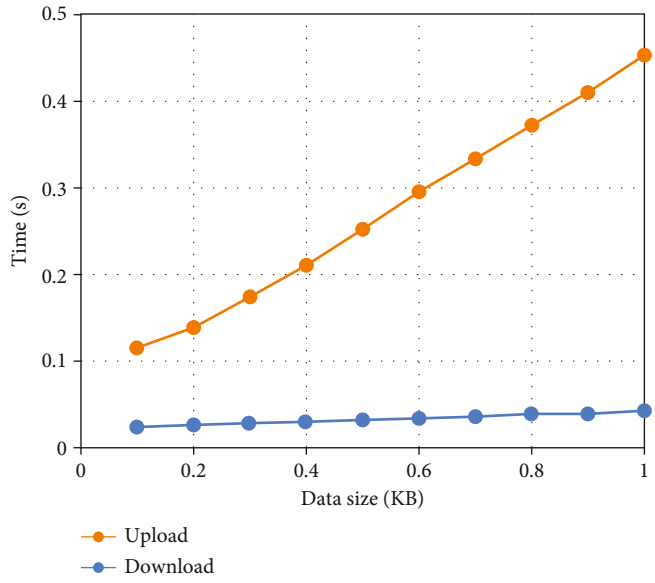

Figure 7: Data transmission time in blockchain.

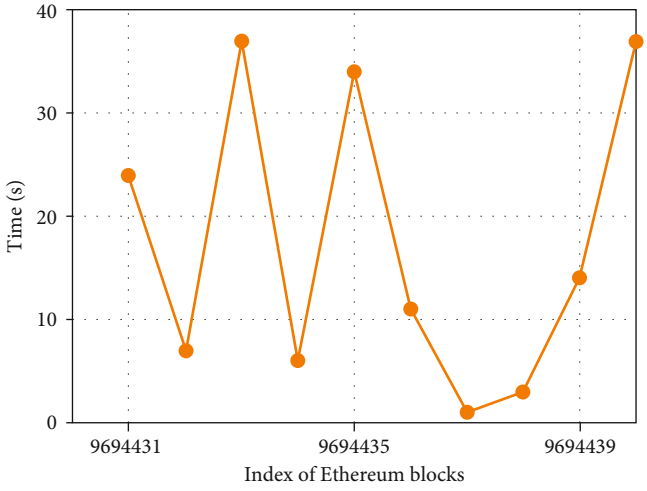

Figure 8: Ten consecutive Ethereum blocks' consensus time.

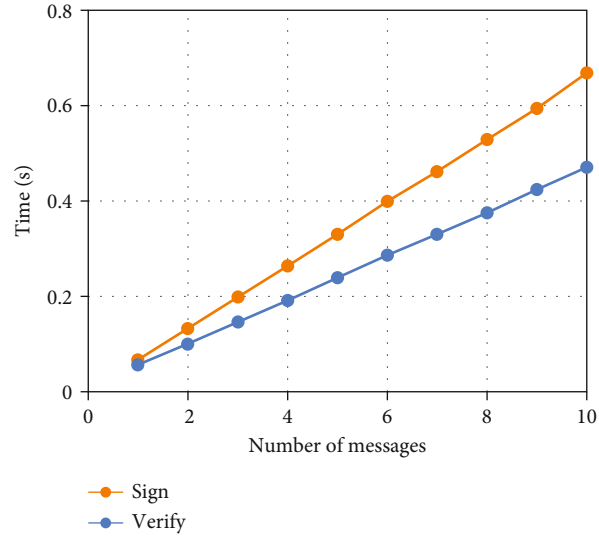

Figure 9: Aggregate signature time consumption.

6.2.3. Time Consumption for Aggregate Signature. The aggregate signature aggregates the digital signatures obtained by signing each user's message. The experimental horizontal coordinate we designed is the number of randomly generated messages, which can also be understood as the number of users. In our proposed method, it is the number of all platoon members. Each member use its secret key x and a message $M_i \in \{0,1\}^*$ to compute $h_i \longleftarrow H(M_i)$, $\sigma_i \longleftarrow h^x$. The signature is σ_i. The randomly generated message is digitally signed, and we can obtain the average time required for a single signature from the repeated experiments. We record the average signing time as T_{sign}. The process of aggregating all the signatures σ_i only takes time σ_{aggr} to compute $\sigma \longleftarrow \prod_{i=1}^{k} = \sigma_i$. This computation does not require too much computing power compared with the power consumed by digital signatures. Let n be the number of messages, and then, the summary of the consumed time is

$$T_{\text{sum}} = n \times T_{\text{sign}} + (n-1) \times T_{\text{aggr}} = \left(T_{\text{sign}} + T_{\text{aggr}}\right) \times n - T_{\text{aggr}}. \tag{4}$$

As shown in Figure 9, the summary time required to digitally sign messages and aggregate the digital signatures increases linearly with the number of messages. The reason for this increase is that it consumes much computational power when digitally signing and aggregating each message.

With the number of original messages generating aggregate signature increasing, the process of verifying the correctness of the aggregate signature also consumes

more time. The number and time have a linear relationship. In the process of aggregate signature verification, we use an aggregate signature σ indexed as before for members, the original messages $M_i \in \{0,1\}^*$, and public keys v_i for members to compute all $h_i \longleftarrow H(M_i)$ and then judge if $e(\sigma, g_2) = \prod_{i=1}^{k} e(h_i, v_i)$. Here, we need to hash each message. Because the message is randomly generated, multiple experiments can be performed to get the average time T_{hash} required for each hash algorithm. In the $\prod_{i=1}^{k} e(h_i, v_i)$ operation before the correctness, the multiplication requires T_{multi}, and the total time required to verify n messages is

$$T_{\text{sum}} = n \times T_{\text{hash}} + (n-1) \times T_{\text{multi}} = (T_{\text{hash}} + T_{\text{multi}}) \times n - T_{\text{multi}}. \tag{5}$$

The performance in the graph is the linear relationship of growth.

6.2.4. Performance of Scheme. Our proposed scheme enables the platoon to pass ETC at one time, and we compare this scheme with the ETC for single vehicle in the actual scenario.

According to the highway ETC standard, the vehicle should comply with the rules of vehicle speed v and distance D_v between two vehicles when passing the ETC. Assume that m cars build a platoon, and the distance between the vehicles inside the platoon is D_p. For n cars with an average length of l, we compare the time T_{platoon} required to pass the ETC after they build a platoon with the time T_{vehicle} required for a single vehicle to pass ETC:

$$\begin{aligned} T_{\text{vehicle}} &= \frac{n \times l + (n-1) \times D_v}{v} = \frac{n \times (l + D_v) - D_v}{v}, \\ T_{\text{platoon}} &= \frac{(n/m) \times \left[m \times l + (m-1) \times D_p\right] + ((n/m) - 1) \times D_v}{v} \\ &= \frac{n \times \left(l + D_p\right) + (n/m) \times \left(D_v - D_p\right) - D_v}{v}. \end{aligned} \tag{6}$$

According to the standard parameters of ETC, let $v = 20$ km/h, $D_v = 35$ m, $D_p = 15$ m, and $l = 5$ m, and assume $m = 8$ cars in a platoon. For different n, the experimental results are shown in Figure 10. The time spent by single vehicles passing through ETC and platoon through ETC increases linearly with the number of vehicles, which is consistent with the calculation result of the formula. When the number of vehicles is the same, it is evident that the platoon consumes less time. As the number of vehicles increases, the time gap between the two lines becomes larger and larger. From this, we can conclude that our scheme saves the time overhead of ETC payments on the highway.

7. Conclusions

In this paper, we propose an efficient highway toll paradigm based on blockchain for opportunistic autonomous vehicle platoon. Vehicles can autonomously build a platform to form a virtual secure communication area relying on blockchain technology. The platoon interacts with RSU to complete the ETC payment preparation phase, and then, the platoon leader leads all platoon members to pass the ETC for finishing the payment quickly. For the bogus information injection and repudiation attacks that may occur between mutually untrusted vehicles, we conducted a detailed security analysis to conclude that our designed protocol can defend against these attacks. Moreover, the experimental results show that the scheme is highly efficient for autonomous vehicles and dramatically reduces the time for ETC deductions.

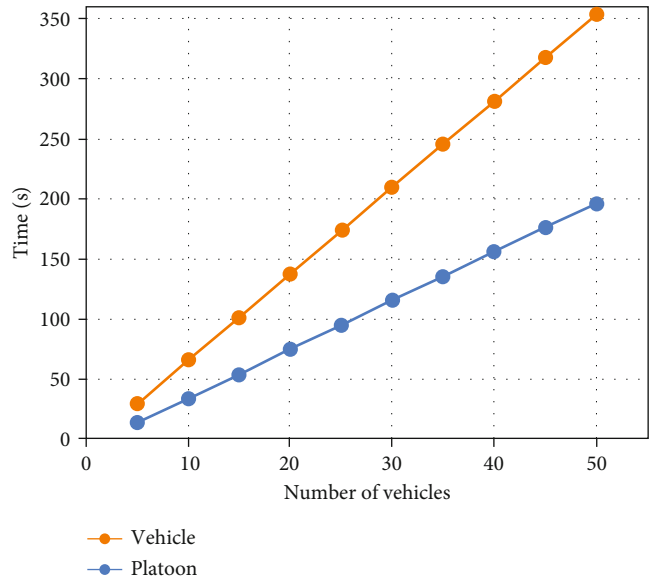

Figure 10: ETC passing time consumption.

Data Availability

The simulation data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the IAF-PP WP5: Design of Security Functionality for V2X Networks Grant for the project of A19D6a0053 by the Agency for Science, Technology and Research, Singapore.

References

[1] A. Driving, *Levels of Driving Automation are Defined in New SAE International Standard J3016: 2014[J]*, SAE International:, Warrendale, PA, USA, 2014.

[2] B. Li, "Stochastic modeling for vehicle platoons (I): Dynamic grouping behavior and online platoon recognition[J]," *Transportation Research Part B: Methodological*, vol. 95, pp. 364–377, 2017.

[3] 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.

[4] B. Besselink, V. Turri, S. H. van de Hoef et al., "Cyber–physical control of road freight transport," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1128–1141, 2016.

[5] V. Sokolov, J. Larson, T. Munson, J. Auld, and D. Karbowski, "Platoon formation maximization through centralized routing and departure time coordination," 2017, http://arxiv.org/abs/1701.01391.

[6] T. Zeng, O. Semiari, W. Saad, and M. Bennis, "Joint communication and control for wireless autonomous vehicular platoon systems," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7907–7922, 2019.

[7] S. Wen and G. Guo, "Sampled-data control for connected vehicles with Markovian switching topologies and communication delay," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2930–2942, 2020.

[8] A. Alam, B. Besselink, V. Turri, J. Mårtensson, and K. H. Johansson, "Heavy-duty vehicle platooning for sustainable freight transportation: a cooperative method to enhance safety and efficiency," *IEEE Control Systems Magazine*, vol. 35, no. 6, pp. 34–56, 2015.

[9] N. Boysen, D. Briskorn, and S. Schwerdfeger, "The identical-path truck platooning problem," *Transportation Research Part B: Methodological*, vol. 109, pp. 26–39, 2018.

[10] S. Gong, J. Shen, and L. Du, "Constrained optimization and distributed computation based car following control of a connected and autonomous vehicle platoon," *Transportation Research Part B: Methodological*, vol. 94, pp. 314–334, 2016.

[11] S. Gong and L. Du, "Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles," *Transportation Research Part B: Methodological*, vol. 116, pp. 25–61, 2018.

[12] C. Wang, S. Gong, A. Zhou, T. Li, and S. Peeta, "Cooperative adaptive cruise control for connected autonomous vehicles by factoring communication-related constraints," *Transportation Research Part C: Emerging Technologies*, vol. 38, pp. 242–262, 2019.

[13] S. Gong, A. Zhou, and S. Peeta, "Cooperative adaptive cruise control for a platoon of connected and autonomous vehicles considering dynamic information flow topology," *Transportation Research Record*, vol. 2673, no. 10, pp. 185–198, 2019.

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

[15] B. Ledbetter, S. Wehunt, M. A. Rahman, and M. H. Manshaei, "LIPs: a protocol for leadership incentives for heterogeneous and dynamic platoons," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, pp. 535–544, Milwaukee, WI, USA, July 2019.

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

[17] Y. Zhang, J. Weng, J. Weng, M. Li, and W. Luo, "Onionchain: towards balancing privacy and traceability of blockchain-based applications," 2019, http://arxiv.org/abs/1909.03367.

[18] 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.

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

[20] 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.

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

[22] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, and C. Z. Gao, "Dynamic fully homomorphic encryption-based Merkle tree for lightweight streaming authenticated data structures," *Journal of Network and Computer Applications*, vol. 107, pp. 113–124, 2018.

[23] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

[24] L. Cheng, Q. Wen, Z. Jin, H. Zhang, and L. Zhou, "Cryptanalysis and improvement of a certificateless aggregate signature scheme," *Information Sciences*, vol. 295, pp. 337–346, 2015.

[25] Z. Ying, M. Ma, and L. Yi, "BAVPM: practical autonomous vehicle platoon management supported by blockchain technique," in *2019 4th International Conference on Intelligent Transportation Engineering (ICITE)*, Singapore, Singapore, September 2019.