Article

An Effective and Secure Key Management Protocol for Message Delivery in Autonomous Vehicular Clouds

Congcong Li ¹, Shouwen Ji ^{1,*}, Xi Zhang ¹, Haiping Wang ², Dongfeng Li ³ and Huiyong Liu ¹

¹ School of Traffic and Transportation, Beijing Jiaotong University, Haidian District, Beijing 100000, China; 14114206@bjtu.edu.cn (C.L.); xizhang@bjtu.edu.cn (X.Z.); 15114198@bjtu.edu.cn (H.L.)

² Department of Research and Development, Beijing Zhonghaiwenda Information Technology Company, Haidian District, Beijing 100000, China; whp@mail.ustc.edu.cn

³ Electronic Transaction Cryptographic Application Group, State Cryptography Administration Office of Security Commercial Code Administration, Fengtai District, Beijing 100000, China; lidongfeng@refinepay.com

* Correspondence: shwji@bjtu.edu.cn; Tel.: +86-139-1078-0769

Received: 19 July 2018; Accepted: 28 August 2018; Published: 31 August 2018

Abstract: Autonomous vehicular clouds, as the combination of cloud computing and conventional vehicular ad hoc networks, will provide abundant resources and services by sharing under-utilized resources of future high-end vehicles such as computing power, storage and internet connectivity. Autonomous vehicular clouds will have significant impact if widely implemented in the intelligent transportation system. However, security and privacy issues are still big challenges in autonomous vehicular clouds. In this paper, after analyzing the particularity of autonomous vehicular clouds, we implement a two-layered architecture, in which vehicles are self-organized without the help of roadside units. Then based on the architecture, we put forward an effective key management protocol to distribute a group key efficiently and also provide the authentication and confidentiality that lots of current secure schemes ignore. In addition, according to the different scenarios and security levels we categorize the way of message transmitting into three kinds. At last, with performance evaluations, the proposed protocol can perform more efficiently than other well-known available schemes.

Keywords: group key management; authentication; certificateless; Chinese Remainder Theorem; vehicular cloud

1. Introduction

With the wide usage of intelligent transportation system (ITS) and the variety of wireless communication technologies, vehicular ad-hoc networks (VANETs), as the important part of ITS, have become of paramount importance in modern traffic management. VANETs provide different services for drivers and allow them to share sensitive traffic information with other drivers such as accident avoidance warnings, weather conditions and state of vehicles, which can improve traffic management efficiency and safety.

However, with the increased shared traffic information and the more types of applications users need such as in-vehicle multimedia entertainment, vehicular social networking and location-based services, a single vehicle has limited computation and storage resources, which leads to low data processing capability [1].

Recently, a few studies have proposed the concept of vehicular cloud computing (VCC) that combines cloud and VANET. VCC is a new paradigm that has a prominent impact on traffic

management and road safety [2] and it has been developed to overcome the drawbacks in VANET. Nowadays, VCC is considered as the key measure to improve and develop ITS [3].

Oraliu et al. [4] first proposed the concept of autonomous vehicular clouds (AVCs). In AVCs, a number of vehicles with abundant resources are viewed as service providers [5]. This is a group of largely autonomous vehicles that contribute their computing, sensing, communication and physical resources to the cloud. Vehicles' resources and the information exchanged from the vehicles with the cloud can be used by other vehicles in decision making [6]. There are two modes in AVCs, zero-infrastructure vehicular clouds and infrastructure-based vehicular clouds. The first mode can provide more stable communication structure due to the high mobility of highway vehicle nodes, which leads to the short of network communication time and the loss of allocated resources. With this in mind, this paper is mainly focused on the zero-infrastructure autonomous vehicular clouds, which few researchers have studied.

With the wide use of AVCs, the significance of security is also on the rise. A more careful analysis reveals that many of the classic security challenges are exacerbated by the characteristic features of AVCs [7]. In addition, it is more complex to distribute keys of vehicles due to the characteristic of zero-infrastructure vehicular clouds. Therefore, an efficient and secure key management protocol, which can manage active groups and provide an authentication as well as confidentiality with less computation and communication overhead, is necessary.

In the study of AVCs, most studies only focus on the security challenges [2,3,7,8], the architecture and traffic flow control [1,9–13], but in the literature, just a few studies have addressed a specific solution to these security challenges in AVCs. Furthermore, among these existing protocols in the active network, most of them have addressed authentication but fail to propose an authentication as well as confidentiality protocol. In this paper, according to the threats in AVCs, we propose an efficient and secure key distribution protocol to ensure the secure communication and the confidentiality of the traffic information in AVCs. The main contributions provided are as follows:

The proposed key management protocol is based on the Chinese Remainder Theory (CRT) and certificateless public key cryptograph (CLPKC) [14], which supports a high secure and efficient authentication and confidentiality in AVCs. The major advantages of the protocol are that updating keys during the users' join and leave operations is performed efficiently. Besides, it solves the certificate management problem in the public key infrastructure (PKI) [15]. With performance evaluations, this proposed protocol gets a promising result from a better trade-off between efficiency and security than other current schemes discussed in the literature. This protocol is considered as a better adaptation in AVCs.

Due to the high mobility of vehicular nodes in the zero-infrastructure scenario, we design a two-layered architecture, in which vehicles self-organize to groups to ensure the better communication stability. Furthermore, in order to reduce the computation overhead at a vehicle side and a cloud server side, the proposed protocol categorizes the traffic information and the way of information transmission into three types, respectively. These features allow our protocol to be performed in various scenarios.

We perform some experiments to measure the energy consumption of the proposed protocol. The experimental results show that our protocol can handle subgroup key management more efficiently than other well-known schemes. Our scheme is lightweight and hence more suitable for AVCs.

The remainder of this paper is organized as follows: Section 2 summarizes the previous works in the literature. The problem statement and the system model are presented in Section 3. We describe our proposed key management protocol in Section 4 and the security analysis about the protocol in Section 5. Section 6 provides the performance evaluation and results of our proposed algorithm with the other current schemes. Section 7 gives a simulator for the proposed protocol. Section 8 gives concluding remarks and suggests some future directions.

2. Related Work

In this section, we provide a brief summary of the related literature addressed the security issues in AVCs and some related popular group key management schemes in VANETs. Yu et al. [1] in 2013 proposed an infrastructure-based architecture for VCs, which included a vehicular cloud, a roadside cloud, and a central cloud. Then they studied cloud resource allocation and virtual machine migration in this cloud-based vehicular network. Hussain et al. [11] in 2014 introduced a multi-layered architecture for VCs and proposed a context-aware dynamic parking service based on the cloud-assisted architecture. Hussain [13] in 2012 proposed a three-architecture framework including vehicular clouds, vehicles using clouds (VuC), and hybrid vehicular clouds (HVC). Ahmed [3] in 2018 proposed a novel privacy-preserving mechanism for VCs. In this paper, the authors focus on the architecture and highlight applications and features in VCs. Also, they explain the challenges for VCs. Finally, the authors present opportunities and future for VCs. However, these aforementioned researches only focused on providing the concept and the architecture for VCs, but the issues of security and privacy have not yet been addressed in the literature.

Aloqaily [16] proposed a quality of experience (QoE) software architecture in vehicular clouds, where QoE requirements are collected via numerous vehicular nodes in the vehicular cloud and re-formulated by a weighted combination of some important factors such as revealed to the Trusted Third Party (TTP). Through this scheme, users can select the best appropriate TTP and service providers (SPs) depending on the QoE-aware values. However, this architecture is multicenter-based, it has to be ensuring the number of TTPs between the users and SPs. The difficulty is to ensure the credible identity of TTPs and to protect from collusion attacks.

Aloqaily, also [17] introduced a scheme that provides continuous vehicular services by predicting the future location of vehicles. This scheme used QoE methodology to provide appropriate services among service providers. However, computing prescient location for vehicles and providing services ahead of time are more time-consuming than providing services using real-time data from sensor nodes.

Sánchez-García [18] in 2015 proposed a secure authentication scheme with applications for urban VANET scenarios using the Spanish eID, which allows authorities to easily and automatically obtain drivers' identities. However, the communication is only from the road authority vehicles (e.g., police vehicles) to other vehicles for requesting a driver to be authenticated in case it is suspected of any road infraction. In addition, the PKI-based authentication mechanisms require additional computation overhead to verify the certificates of others.

Rajeshwari [7] in 2016 discussed that security and privacy are the two major challenges in AVCs and it is divided into three subsets in this paper: security threats, security requirements, and security challenges. The paper classified the architecture of AVCs into three layers: networks, the wireless communication channel and cloud computing. Also, it considered that PKI is one of the most important key management in AVCs. However, PKI will have problems in certification management and keys revocation in AVCs.

Wang [19] in 2016 presented a secure and privacy-preserving navigation scheme for fog-based VANETs. This scheme based on a message-locked encryption (MLE) and randomized convergent encryption (RCE) supports authentication, confidentiality and conditional privacy preservation. However, this scheme only accounts for the location matching without considering content matching. The scheme leads to additional communication and computational costs, because that the user will decrypt all the returned cipher texts included in the route and has to find the point of interest.

Otoun [20] in 2018 proposed an adaptively supervised and intrusion-aware data aggregation for wireless sensor clusters in critical infrastructures. The proposed scheme undertook the intrusion problem by using adaptation strategy to detect dynamically known and unknown intrusions. And it demonstrated that the proposed method performs with 99% detection rate and 99.80% accuracy in the presence of known and unknown malicious behavior in the WSN. However, it only focuses on the detection strategy without active defense strategy.

According to threats in AVCs, some literature proposed related methods to ensure the security in AVCs. The secure protocols listed below are based on the certification.

Yan et al. [7] in 2013 analyzed a variety of security challenges and potential privacy threats that are specific to VCs. Then this paper proposed a geographic location-based security mechanism to ensure security for VCs. The scheme divides the city map into a number of cells with a virtual machine. However, the vehicle could not decrypt the messages if it is not in the decryption region. Therefore, the scheme has a limited scalability of communication in VCs. Also, the scheme is based on the PKI, which may cause an additional communication overhead for managing certifications. At last, the key revocation of the scheme is using certificate revocation lists (CRLs), which will occupies the large amounts of storage.

Lin et al. [21] in 2008 proposed a timed efficient and secure vehicular communication (TSVC) scheme using hash chains to protect the privacy of users. In this scheme, it utilizes the group signature without helping of infrastructure to verify messages. Due to the short message authentication code (MAC) attached to each packet, the communication and computation overhead are significantly reduced compared with other current public key based authentication schemes. However, this scheme is based on the certification and each vehicle is preloaded with many anonymous public/private key pairs and their corresponding public key certificates at the initialization stage, which occupies more storage of vehicles than other certificationless schemes. Moreover, when vehicle members frequently change from a group, the leader vehicle of the group broadcasts its hash chain much more frequently, which will induce the increased message loss ratio. Also, the scheme fails to propose a confidentiality services for users.

Because of the certificate management problem, some other schemes proposed ID based authentication for vehicular-network applications.

Abdulrahman et al. [5] in 2017 proposed a lightweight privacy-preserving scheme, which provided confidentiality and integrity in AVCs. This scheme supports RSU-based architecture and the RSU, acts as the proxy ID, resigncrypts the messages from a certificate authority (CA) server. However, because of the high mobility of vehicles in AVCs, it is time-consuming for RSUs to resigncrypt messages. Besides, the information on the scheme is one-way transmission from the CA server to vehicles and it does not support vehicle-to-vehicle communication. Furthermore, the CA server generates the secret key for every user and hence it cannot resist insider attacks. Finally, this method is based on bilinear pairing and the relative computation overhead of a pairing operation is approximately 20 times higher than that of an elliptic curve scalar multiplication [22].

Chim [23] in 2011 proposed an ID-based authentication scheme based on the bilinear pairings with lower communication costs for secure V-to-I communications. However, Horng et al. [22] in 2013 found that Chim's scheme was vulnerable to impersonation attacks. Horng et al. [24] provided a secure scheme that can achieve the security and privacy requirements, and overcame the weaknesses of the scheme in [23]. However, the computation overhead of one pairing operations is at least three times higher than that of a one point multiplication operation [25]. Hence these schemes based on bilinear pairings are not suitable for rapidly changing networks. Also, these schemes cannot ensure the confidentiality of messages, which could not support privacy messages communication.

Park [26] in 2018 proposed a trajectory-based message delivery for cloud-assisted VANETs. The vehicles can share their trajectory messages to specified socialspot roadside units through the center. The strategy adopted a certificateless proxy re-encryption (CL-PRE) scheme and an ID-based signature (ID-SIG) scheme, with which it can get better security for user's privacy messages. However, this scheme fails to support the conditional privacy preservation. The TA cannot trace malicious vehicles and obtain their real identities, which are not stored in the TA to protect from table-stealing attacks. Furthermore, in the phase of trajectory sharing on the cloud, when one vehicle send the message $Tm_d = \{pid_{d0}, C, ts, S_d\}$, and id_{d0} , a unchanged PID, is easy for malicious vehicles to link messages from the same vehicle. In addition, the relative computation overhead of a pairing operation is higher than that of an elliptic curve scalar multiplication.

He et al. [27] in 2015 proposed an ID-based scheme using the Elliptic Curve Cryptography (ECC), which was more efficient than previously proposed authentication schemes. However, this scheme heavily relies on a tamper-proof hardware device. Lo et al. [28] in 2016 proposed a faster ID-based scheme based on ECC and the scheme supported batch verification. However, it is significantly dependent on secure communication and it has high costs and limited scalability. Also, these two schemes are only focus on authenticity but the confidentiality.

Yang [29] in 2018 proposed an authentication scheme based on certificateless signature with message recovery. The scheme supports conditional privacy, batch verification and unlinkability in VNENTs. The evaluation results indicate that the scheme achieves low computation and communication costs without the bilinear pairing and map-to-point hash operations. However, this scheme is a one-way communication only from vehicles to RSUs, where vehicles cannot access the traffic information sent from the center. This transmission mechanism affects the function of information sharing in VENETs.

In the following, we will discuss some of the current group key management methods, which not only support integrity but confidentiality.

Lim [30] in 2016 proposed an efficient group signature scheme based on certification in VANETs. Each RSU acts as a leader of one group, is responsible for the distribution and updating of pseudo identities (IDs), original IDs, certificates and shared secret keys. However, according to the driving speed of individual vehicles, especially in the highway scenario, the distance between one mobile vehicle and a static RSU is changed quickly. Therefore, it is time consuming to initiate the key establishment process frequently when the vehicle enters a communication region covered by a new RSU. Furthermore, in this infrastructure-based architecture, if a vehicle wants to download some huge documents, it will fail to finish downloading when moving out from the communication range of the RSU. Also, the scheme does not provide the group key updating process when one member leaves or joins a group, which is not supporting the backward and forward secrecy. When one vehicle leaves or joins the region of a RSU, it can also access the information from the RSU using the group key K_{G_i} .

Vijayakumawe [31] in 2016 proposed a dual group key management scheme for VANETs. In this scheme, the vehicle users are categorized into three groups, which have different authorities to access the service. With the distribution and updating of a group key via Chinese Remainder Theorem (CRT), this scheme can get a more computationally efficient result compared with other related schemes. However, the information transmission is in a one-way manner, in which the scheme does not refer to how Secondary Users (SUs) or Primary Users (PUs) delivers messages to the TA. Furthermore, it is not efficient for message transmission. If there is no PUs around SUs, SUs could not obtain traffic information in time. In addition, with the group secrete key of SUs, PUs can learn about any of information of SUs. At last, the scheme is lacking of privacy preservation. When the PUs broadcast the message to SUs by $E_{k_{sug}}(ID_v \parallel message) \parallel E_{TA-Pvt}(AC \parallel ID_v \parallel TS_3 \parallel Lifetime)$, the identity of this PU ID_v is exposed to the public.

Zheng et al. [32] in 2007 proposed two centralized group key management protocols (CRGK and FCRGK) based on the CRT. The proposed approach minimizes the storage complexity at the TA side and the number of broadcast messages at the user side. However, the computation complexity is high at the key distribution aspect.

Zhou [33] in 2009 designed a group key distribution algorithm based on the CRT. The main advantages are that it reduces the key server's computation complexity of the group key distribution and when the number of group members increase to a certain number, the computing time of the key server will decrease. However, this also increases the workload of finding the root IDs in the group member subtrees and a common group key by congruent equations.

Seo [34] in 2015 proposed a certificateless-effective key management (CL-EKM) protocol in dynamic wireless sensor networks (WSNs). It is noted that it can significantly reduce the communication overhead by CL-EKM based on the cluster structure. This scheme supports efficient

key updating when a node leaves or joins a cluster and ensures forward and backward key secrecy. However, the method of updating cluster key is more time consuming. The group leader informs every member of new keys by pairwise session keys of its cluster.

Compared with the current authentication and group key management schemes mentioned in the literature, in this paper, we propose a key management protocol that is superior to others in many ways. First, the protocol is based on the certificateless public key cryptograph (CLPKC) using ECC, which can solve the certificate management problem in the PKI and the key escrow problem in the identity-based encryption (IBE) with lower computational overhead. Furthermore, the protocol provides authenticity plus confidentiality for communication among vehicles. Moreover, according to the different security levels we categorize keys into four kinds and traffic information into three kinds. In order to minimize the computation overhead of key distribution and management, we use the CRT-based key management approach. With performance evaluations, the computational cost in the proposed protocol is lower than other related schemes.

3. Proposed Vehicular Networks Architecture

3.1. Problem Statement and Solution Objectives

3.1.1. Time-Consumption

We consider two scenarios here. For urban roads in a city block, a number of vehicles are prone to co-located when a traffic-related event or traffic jam has occurred, which leads to a low average speed of vehicles. Multiple vehicles which observe the same phenomena will send repetitive traffic-related messages to the cloud sever which can result in the increased burden for the cloud server. Therefore, the messages should be analyzed, verified and discarded if it redundant before sending to the cloud server.

3.1.2. Cost-Effectiveness

On highways, vehicles are distributed non-uniformly, and the high mobility of vehicles leads to short-time communication in the network. Therefore, the communication range between mobile vehicles and fixed infrastructure such as RSUs, acting as gateway to link vehicles and cloud servers, is limited. Furthermore, the RSUs are deployed in the highways, unattended environments, so the cost of constructing infrastructure is high. From an implementation perspective, the designed protocol should ensure the stability of the communication quality and the low cost high efficiency.

3.1.3. Verification

When a vehicle encounters problems such as accidents, bad road conditions due to bad weather or traffic jams, etc., it needs to propagate that information to other nearby vehicles and the transportation centers in appropriate regions. If an adversary inserts malicious messages or modifies messages to the network, this will lead to traffic chaos or even accidents. Therefore, the authentication of the messages should be ensured.

3.1.4. Anonymity

The anonymity of drivers should be preserved to protect the identities of drivers from being revealed by adversaries. Also, the information sent from the vehicles should not be traced by other vehicles.

3.1.5. Confidentiality

Some information about privacy of drivers should be protected from being eavesdropped. Thus, the confidentiality of messages should be ensured.

3.1.6. Light Weight

Due to the high mobility of vehicles and the limited computation power of on board units (OBUs), a lightweight key management protocol with fast authentication and encryption/decryption should be designed.

In this paper, we present an efficient key management protocol, which addresses and solves the aforementioned problems.

3.2. Cloud Model

According to the two scenarios in city blocks and highways mentioned above, we adopt a zero-infrastructure with two-layer architecture to better suitable for the AVCs. The two-layered architecture includes the autonomous vehicular cloud (AVC) and the central cloud (CC):

- *Autonomous Vehicular Cloud:* A mobile cloud consists of a large number of mobile sensor nodes played by autonomous vehicles and a set of aggregation-and-forwarding nodes, namely subgroup leaders (GLs). Each GL aggregates the data from the other vehicle nodes in its subgroup and forwards to the CC. The vehicles provide under-utilized resources and in return they receive the welfare from the central cloud for services [35]. Vehicles in the vicinal communication range select the GL by the method [36] which acts as the gateways. All vehicles in VCs are divided into many subgroups that are administered by their respective GLs. The vehicles can access the resources from cloud servers by vehicle to cloud communication (V2C) and other vehicles by vehicle to vehicle (V2V) communication. As a self-organized group of cooperative vehicles, they can lengthen the communication durability within its own communication range. The decentralized architecture can make less cost of the central cloud and reach an effective implement and key delivery.
- *Central Cloud:* A central cloud has many more resources and is mainly used for registration, complicated computation, massive data storage, and global decisions [1]. The central cloud provides services for vehicles. Here, mobile edge computing (MEC) implemented at the base stations (BS) is useful in reducing latency and improving localized user experience [37]. Delay-sensitive services are executed and delivered much faster due to the close proximity of end users to the MEC infrastructure [38].
- *Private Key Generator (PKG):* The PKG included in the CC is a trusted party that generates partial private keys to users. The PKG is fully trusted by all parties in the network and cannot be compromised.

This cloud architecture has several essential advantages. First, energy-sufficient vehicles can act as ideal active candidates to provide their storage, computational ability and sensing capabilities on the move. Second, this architecture is especially adapted to the highway scenario, which can cut down the expenditure of building fixed infrastructure. Furthermore, the average speed of vehicles is high. The vehicles form a subgroup to avoid communication breakage and it is helpful to enlarge the communication range when accessing the data by V2V rather than the fixed infrastructure, so the self-organized group structure is beneficial for the continuity and reliability of accessing cloud services. Last but not least, it is more convenient and secure for the CC and GLs to manage secret keys. Further details on the protocol and security proof are elaborated in the next section.

4. Proposed Protocol

In this section, we present an effective and secure key management protocol for autonomous vehicle clouds. In order to improve the security and reduce the energy consumption of communication, the proposed protocol categorizes the transmitting ways into three kinds according to the different security levels. There are four kinds of keys for different message delivery: a group key r within the range of all authorized vehicular nodes, a subgroup key GK , a secret key d_2 between the PKG and a vehicle, and a secret key k between a group leader and a vehicle.

4.1. Types of Keys

The functions of the different types of keys are shown in Table 1.

- *The Secret Key between the PKG and a Vehicle:* Each vehicle shares a unique individual key d_2 with the PKG. For example, if a vehicle wants to transmit the sensitive data to the CC directly bypassing the GL, or if it fails to communicate with the GLs, the node can encrypt the information using the secret key and send it to the CC. In addition, it is convenient for the CC to calculate the group key via d_2 .
- *Group Key:* We consider all the authorized vehicles as one large group, in which all registered vehicles share one secret key. The group key is mainly used for broadcasting traffic messages without obtaining by other illegal vehicles, and also it can exclude the compromised node from the group easily through changing the group key. Only the CC can update the key when a vehicle leaves or joins the group.
- *Subgroup Key:* In the cluster-based structure, the pool of members is categorized into small groups called subgroups, which can minimize the computation complexity in the key updating phase and achieve a secure key delivery. All vehicles in a subgroup share one key, named subgroup key, which is mainly used for securing broadcast messages, such as sensitive commands or the change of member status in one subgroup. Only the GL can update the subgroup key when the member joins or leaves the subgroup.
- *Secret Key between the GL and the Vehicle:* The function of constructing the secret key between the GL and the vehicle is to calculate the subgroup key for them. It is a secure way to deliver the secret key k by key agreement instead of generating it by the GL itself.

Table 1. Functions of keys.

Keys	Functions
The secret key between a vehicle and the PKG d_2	<ul style="list-style-type: none"> • As a partial secret key in authentication process. • Calculating the group key. • An important secret token to trace the real identity of the vehicle for the CC.
Group key	<ul style="list-style-type: none"> • Broadcasting traffic messages. • Excluding the compromised vehicle nodes.
The pairwise key between the GL and the vehicle k	<ul style="list-style-type: none"> • Calculating the subgroup key.
Subgroup key	<ul style="list-style-type: none"> • Broadcasting subgroup-sharing messages. • Fast authentication for subgroup members.

4.2. Types of Information

In this section, the proposed protocol divides the information into three types:

- *Public traffic information:* This kind of traffic information collected by vehicular nodes and analyzed by the central cloud aims to help vehicles take intelligent decisions and enhance security and quality of driving. These traffic messages based on related services and applications include real time traffic information, intelligent decision hint, localizing events and emergency warning etc.
- *Private custom information:* This type of information is related to the privacy of drivers, such as parking space booking, infotainment, electronic logging, service spot detection, specific

route navigation and some other sensitive data. The different service requirements should be encrypted to protect from being eavesdropped.

- *Group sharing information:* This type of information used to be shared with all members in one subgroup, such as the private change of member status, updating subgroup keys and subgroup services etc. This kind of information should be acquired by the subgroup members rather than others.

The different kinds of information are summarized in Table 2.

Table 2. Functions of three kinds of information.

	Contents	Security level	Keys
Public traffic information	Real time traffic information, Intelligent decision hint, Localizing events, Emergency warning etc. Parking space,	Access with all authorized vehicles	Group key, Subgroup key.
Private custom information	Infotainment, Electronic logging, vService spot detection, Specific route navigation etc.	Encryption Only access with one specified vehicle	Subgroup key, Secret key d_2 .
Group sharing information	Private change of member status, Updating subgroup keys, Subgroup services etc.	Encryption Access with all members in one subgroup.	Subgroup key, Secret key d_2 .

4.3. Types of Transmitting

In order to strengthen the security of information communication in the autonomous network, this proposed protocol considers three ways of information transmission according to the different security level of information as shown in Figures 1 and 2.

Figure 1. Transmission path of Public traffic information.

Figure 2. Two kinds of transmission path: (a) Subgroup-sharing information; (b) Private custom information.

In the first scenario (seen in Figure 1), the vehicles share the public traffic information in real time. The difference from other protocols is that the GL classifies and filters the messages after receiving them from subgroup members and then transfers them to the central cloud. With the help of the framework based on the efficient vertical-handover operation (VHO) and service-level-agreements (SLAs) scheme [39], the mobile nodes, especially GLs, can access different networks with stable communication and low service delay.

The benefit is that this can reduce the redundant and repetitive traffic messages and achieve cost reductions, especially in intersections or the spot of an accident where a lot of vehicles are co-locating for the red light or traffic events. Then, the central clouds broadcast the messages to the authorized members directly.

The second scenario (seen in Figure 2a) shows that the GL sends the requirement to the central cloud without being known by other vehicles during the subgroup key updating process. Then the central cloud feeds back the related information to the GL.

In the third scenario (seen in Figure 2b), the GL transfers the private custom information to the CC, and then it returns the service information to the vehicle directly through the end-to-end way bypass the GL.

4.4. The Details of the Protocol

In this subsection, we present the basic idea behind our protocol, which is built on the above CLPKC scheme proposed in [40]. The protocol consists of the following seven phases: system setup, registration, subgroup formation and pairwise key generation, subgroup key generation, group key generation, data transmitting and key updating. The notations used in this paper are listed in Table 3.

Table 3. List of notations.

Symbol	Descriptions	Symbol	Descriptions
n	A k-bit prime number	d_1	A secret key of a user
F_n	A finite field with n elements	d_2	The secret key between a vehicle and the PKG
$E(F_n)$	An Elliptic Curve over a finite field $F_n, y^2 = x^3 + ax + b \bmod n, a, b, x, y \in F_n$	P_1	A public key of a user
G	An additive group with the order q	k	The pairwise key between the GL and the vehicle
q	The order of the group G	GK	A subgroup key
P	The point generator of the group G_q	P_{pub}	A public key of the PKG
PID	The pseudo identity of a user	t	A timestamp
ID	The real identity of a user	$\oplus$	Exclusive-OR operation
r	A group key	$\parallel$	Message concatenation operation
s	A private key of the PKG		

4.4.1. System Setup

The PKG generates system parameters and registers the vehicle via running following steps. First, the PKG takes a k-bit *prime* security parameter n and generates the tuple $\{F_n, E(F_n), G_q, P\}$. The PKG chooses a random number $s \in Z_q^*$ as its private key and computes $P_{pub} = s \cdot P$. Then, the PKG determines three one way hash functions: $h_0 : G_q^* \rightarrow Z_q^*$, $h_1 : G_q^2 \times \{0, 1\}^* \rightarrow Z_q^*$, $h_2 : G_q^2 \times \{0, 1\}^* \times G_q \times \{0, 1\}^* \rightarrow Z_q^*$. At last, the PKG returns system parameters $Z = \{F_n, E(F_n), G_q, P, P_{PKG}, h_0, h_1, h_2\}$ and keep s secret.

4.4.2. Registration

In this phase, we describe the protocol for registering vehicles. The user who wants to access services must disclose his valid credentials such as ID number or driving license number to the PKG. Note that the real identity is registered in the PKG offline and can uniquely identify the user. Then, the user picks a random number $d_1 \in Z_q^*$ as his secret key and computes the public key $P_1 = d_1 \cdot P$. The user encrypts $\{ID, P_1\}$ using the PKG's public key and sends it to the PKG. Upon receiving the $\{ID, P_1\}$, the PKG generates pseudo identities and partial secret keys for the user as below:

$$PID = (K + E_{P_{PKG}}(ID)) \cdot P \quad (1)$$

$$d_2 = (K + E_{P_{PKG}}(ID)) + h_1(P_1, PID, T) \times s \bmod q, \quad (2)$$

where $K \in Z_q^*$ is a random number and T is the period of validity of the PID . The PKG sends $\{PID, P_1, d_2, T\}$ back to the user. Here, there is a one-to-one correspondence between the $PIDs$ and the d_2 s and they are sent by sequence. The PKG deletes ID to protect user's privacy from insider attacks. Upon receiving the partial secret keys, the user checks the authenticity of $\{PID, P_1, d_2, T\}$ via running:

$$d_2 \cdot P = PID + h_1(P_1, PID, T) \cdot P_{pub}. \quad (3)$$

If the equations hold, it means that these $\{P_2, d_2\}$ are generated by the PKG and the user can be deemed a legal one in VCs. Otherwise, the user is rejected. Then the user stores the secret information $\{PID, d_2, P_1, d_1\}$ into the OBU that is a tamper-proof device.

4.4.3. Subgroup Formation and Pairwise Key Generation

The proposed subgroup formation utilizes the scheme in [36], which selects a group leader according to the direction and relative speed of vehicles in the vicinity. The scheme ensures a stable subgroup with less frequency of re-affiliation.

After the subgroup generation, the process of GL establishing the pairwise key with its group members is as follows:

We assume that there is a GL GL_i and one of its member V_j .

First, the GL_i chooses a random number d_{3GL} and one of PID_{GLs} , then calculates:

$$PT_{GL} = d_{3GL} \cdot P \quad (4)$$

$$v_{GL} = d_{3GL} + d_{2GLi} + d_{1GL} \times h_2(PID_{GLi}, P_{1GL}, M_{join}, PT_{GL}, t) \bmod q, \quad (5)$$

where t is a timestamp of the vehicle's system. Then the GL_i broadcasts inquiry messages $\{M_{join}, v_{GL}, P_{1GL}, PID_{GL}, PT_{GL}, T_i, t\}$ to its members within the range of communication.

The vehicle V_j in the subgroup receiving the GL_i 's joining messages at time t' verifies the messages as follows:

- Validate the freshness of t' by checking $t' - t \leq \Delta T$, where ΔT indicates the valid time interval.
- V_j checks the expiration time T of PID .
- V_j checks the equation:

$$v_{GL} \cdot P = PT_{GL} + PID_{GLi} + h_1(P_{1GL}, PID_{GLi}, T_{GL}) \cdot P_{pub} + P_{1GL} \cdot h_2(PID_{GLi}, P_{1GL}, M_{join}, PT_{GL}, t) \quad (6)$$

If it holds, V_j accepts the M_{join} and calculates:

$$PT_j = d_{3j} \cdot P \quad (7)$$

$$v_j = d_{3j} + d_{2j} + d_{1j} \times h_2(PID_{ji}, P_{1j}, M_{agree}, PT_j, t) \bmod q, \quad (8)$$

then it responds GL_i with $\{M_{agree}, v_j, P_{1j}, PID_{ji}, PT_j, T_j, t\}$, else outputs "invalid".

Upon receiving the message, GL_i verifies v_j as follows:

$$v_j \cdot P = PT_j + PID_{ji} + h_1(P_{1j}, PID_{ji}, T_j) \cdot P_{pub} + P_{1j} \cdot h_2(PID_{ji}, P_{1j}, M_{agree}, PT_j, t) \quad (9)$$

If it holds, then GL_i computes the pairwise key between V_j and itself:

$$K_{GL-j} = h_0(d_{3GL} \cdot (P_{1j} + PID_{ji} + h_1(P_{1j}, PID_{ji}, T_j) \cdot P_{pub})) \quad (10)$$

Furthermore, GL_i selects a random number l and sends $(h_{K_{GL-j}}(l \parallel t), l, t, PID_{ji})$ to V_j . Here, $h_{K_{GL-j}}(l \parallel t)$ is an encrypted digest called hashed message authentication code (HMAC), which is viewed as a hash function and encrypted by the session key K_{GL-j} shared between the two entities.

When V_j receives the messages, it computes:

$$K'_{GL-j} = h_0((d_{1j} + d_{2j}) \cdot PT_{GL}) \quad (11)$$

and it checks whether the equation holds:

$$h_{K_{GL-j}}(l \parallel t) = h_{K'_{GL-j}}(l \parallel t) \quad (12)$$

If it holds, that means GL_i and V_j share one secret key K_{GL-j} without being exposed by others.

The steps of the pairwise key generation phase are depicted in Figure 3.

Figure 3. Pairwise keys generation process.

4.4.4. Subgroup Key Generation

After building subgroups, to distribute a common subgroup key for a group member is a GLs responsibility. In this protocol, we use CRT [41] to efficiently manage keys in the subgroup during the users' join and leave operations.

We take the subgroup G_i , its group leader GL_i and one vehicle V_j of group members for an example. G_i has n members. The process of subgroup key building is as follows:

Let $k_1, k_2, \dots, k_n$ are the pairwise keys shared between GL_i and its members.

GL_i computes:

- $M = \prod_{x=1}^n (k_x)$
- $m_x = \frac{M}{k_x}$
- Find y_x such that $y_x \cdot m_x = 1 \bmod k_x$
- Select a random element GK_i as the subgroup key and compute:

$$GKS_i = GK_i \times \sum_{x=1}^n y_x \times m_x \quad (13)$$

Then GL_i multicasts GKS to its subgroup members in public.

Upon receiving GKS value at the GL_i side, the legal vehicle such as V_j can obtain the subgroup key GK_i by doing one modulo division operation as follows:

$$GK_i = GKS_i \bmod k_x \quad (1 \leq x \leq n) \quad (14)$$

V_j obtains the subgroup key GK_i by modulo its pairwise secret key shared with GK_i .

4.4.5. Group Key Generation

Every legal vehicle accessing services would be distributed a group key shared by the PKG. As mentioned above, the group key is mainly used for broadcasting traffic messages while protecting them from other illegal vehicles. The PKG is responsible for the distribution of group key based on the secret key that is shared between the PKG and the legal vehicle. The process is as follows: the PKG uses d_{2s} stored in the PKG during registering phase to produce group keys directly, which seems as a convenient and secure way. Here, the d_{2x} is chosen by sequence. For example, if there are n d_{2s} and $PIDs$ of one user, to identify the right secret key for the user, the PKG chooses d_{2_1} at first time, d_{2_2} at second time and the rest can be done in the same manner:

We assume that there are m legitimate members.

The PKG computes:

- $N = \prod_{x=1}^m (d_{2x})$
- $n_x = \frac{N}{d_{2x}}$
- Find y_x such that $y_x \cdot n_x = 1 \bmod d_{2x}$
- Select a random element r as the subgroup key and compute:

$$rs = r \times \sum_{x=1}^m y_x \times n_x. \quad (15)$$

Then the PKG multicasts rs in public.

When receiving rs , the user that has d_{2s} can recover the value of group key r by doing a modulo division operation. Note that the user chooses d_{2i} by the sequence:

$$r = rs \bmod d_{2x} \quad (1 \leq x \leq m). \quad (16)$$

4.4.6. Data Transmission

In this subsection, we elaborate on the secure communication based on three types of transmission in Section 4.3:

1. In the first scenario, the vehicle V_j shares traffic public information with the subgroup members and it sends the messages within the communication range:

$$\{E_{PKG}(d_{2j} \parallel P_{1j} \parallel PID_{ji} \parallel M_{public} \parallel t), h_{GK_i}(M_{public} \parallel t), M_{public}, t\} \quad (17)$$

Note that it just needs one HMAC operation to verify this message, which is faster and more efficient than the protocols in [24,30,34]. Furthermore, it can support both unlinkability and traceability by the PKG through $E_{PKG}(d_{2j} \parallel P_{1j} \parallel PID_{ji} \parallel M_{public} \parallel t)$, the further details of which will be provided in details in the following sections.

The member such as V_k in the same subgroup can obtain the credible traffic messages easily. As the leader group such as GL_i , is responsible for filtering and transferring messages to the CC instead of sending many same messages by different vehicles. GL_i transmits

$\{E_{P_{PKG}}(d_{2j} \parallel P_{1j} \parallel PID_{ji} \parallel M_{public} \parallel t), M_{public}, t\}$ to the PKG. This method can reduce the communication and computational overhead at central cloud side significantly.

After receiving a variety of traffic messages, the CC will make some critical decisions and adjustment, then it broadcast relative traffic messages to vehicles directly to help them enhance driving safety and increase traffic efficiency:

$$\{Sign_s(M_{traffic} \parallel t), E_r(M_{traffic} \parallel t), t\} \quad (18)$$

The legal vehicles can obtain $M_{traffic}$ via decrypting $E_r(M_{traffic} \parallel t)$ using group key r and verify the messages via P_{PKG} .

2. In the second scenario, the vehicle V_j wants to share the information within the subgroup G_i .

$$\{E_{P_{PKG}}(d_{2j} \parallel P_{1j} \parallel PID_{ji} \parallel M_{subgroup} \parallel t), E_{GK_i}(M_{subgroup} \parallel t), t\} \quad (19)$$

The members in the G_i can obtain the information $M_{subgroup}$ by decrypting $E_{GK_i}(M_{subgroup} \parallel t)$ with the subgroup key GK_i .

If the information is about updating the subgroup key or private changing of member status, which needs to be submitted to the CC, GL_i transfer the message to the CC. The CC sends back the messages after processing it.

For convenience, we set $M' = E_{P_{PKG}}(d_{2j} \parallel P_{1j} \parallel PID_{ji} \parallel M_{subgroup} \parallel t)$, the message as follows:

$$\{E_{P_{PKG}}(M' \parallel P_{1GL} \parallel d_{2GL_i} \parallel t' \parallel PID_{GL_i}), PID_{GL_i}, t'\} \quad (20)$$

If it is about updating subgroup keys, GL_i can send M_{update} to the CC like this:

$$\{E_{P_{PKG}}(M_{update} \parallel P_{1GL} \parallel d_{2GL_i} \parallel t' \parallel PID_{GL_i}), PID_{GL_i}, t'\} \quad (21)$$

where t' is the new timestamp of GL_i .

When the CC receives the message, it decrypts with master key s . Then the CC sends back to GL_i with $\{Sign_s(M_{back} \parallel t), E_{d_{2GL_i}}(M_{back} \parallel t), PID_{GL_i}, t\}$.

GL_i can obtain the message by decrypting it with secret key d_{2GL_i} and verify it by P_{PKG} , and then GL_i sending the secret information to multiple receivers in the subgroup:

$$\{Sign_s(M_{back} \parallel t), E_{GK_i}(M_{back} \parallel t'), t, t'\} \quad (22)$$

where t' is the new timestamp of GL_i .

The vehicles in the subgroup can obtain the messages via decrypting it by the subgroup key GK_i .

3. In the last scenario, when one vehicle sends information about user's privacy, like private custom information, the vehicle V_j sends information as follows:

$$\{E_{P_{PKG}}(d_{2j} \parallel P_{1j} \parallel PID_{ji} \parallel M_{private} \parallel t), h_{GK_i}(PID_{ji} \parallel t), PID_{ji}, t\} \quad (23)$$

GL_i transmits the messages to the CC. Note that GL_i could not obtain $M_{private}$ but can verify this identity of the member by $h_{GK_i}(PID_{ji} \parallel t)$.

The CC sends back to the V_j directly after processing $M_{private}$:

$$\{E_{d_{2j}}(M_{back} \parallel t'), PID_{ji}, t'\} \quad (24)$$

The steps of the data-transmitting phase are depicted in Figure 4.

Figure 4. The different scenarios of data transmitting.

4.4.7. Key Updating

In VCs, an efficient group key updating operation is indispensable because the members in the dynamic subgroups change frequently. When a vehicle leaves or joins a subgroup, the group leader will update the subgroup key to ensure forward and backward secrecy. The proposed protocol provides an efficient way to update keys with less computation overhead than the schemes in [24,30,34].

For example, when a vehicle V_j of subgroup G_i leaves the subgroup, it may notify the group leader GL_i and the group leader decides to revoke the vehicle V_j . GL_i subtracts $y_j \times m_j$ from $\sum_{x=1}^n y_x \times m_x$ and then selects a new subgroup key GK_i' :

$$GKS_i' = GK_i' \times \left(\sum_{x=1}^n y_x \times m_x - y_j \times m_j \right) \quad (25)$$

The updated subgroup key value GKS'_i is sent as a broadcast message to all the members in G_i . The existing member such as V_m can obtain GK'_i by doing only one operation:

$$GK'_i = GKS'_i \bmod k_m \quad (26)$$

Note that V_j could not calculate the new subgroup key when receives the GKS'_i , because its secret value is not included in $\sum_{x=1}^n y_x \times m_x$.

Similarly, if one vehicle V_s wants to join the subgroup G_i , after sending the request message including its signature, GL_i verifies the message and computes the pairwise keys k_s with V_s described in Section 4.4.3.

Then GL_i sends the request of computing subgroup key to the CC with the form of second data-transmitting scenario mentioned in Section 4.4.6. After obtaining relative parameters, GL_i performs one addition operation and selects a new subgroup key GK'_i :

$$GKS'_i = GK'_i \times \left(\sum_{x=1}^n y_x \times m_x + y_s \times m_s \right) \quad (27)$$

The newly joined V_s finds the newly updated group key via computing.

$$GK'_i = GKS'_i \bmod k_s \quad (28)$$

5. Security Analysis

First, we briefly review the security of CLPKC [40]. Later, we discuss how the proposed key management protocol based on CLPKC achieves the security goals. The CLPKC scheme supports unforgeability against adaptive chosen messages attacks, message authentication and integrity, which is discussed in detail in [40]. The other security analyses are given as follows.

5.1. Resistance to Replay Attack

In a replay attack, the malicious vehicle resends previously received messages back to the AVCs. In our protocol, to protect our system from replay attacks, all valid messages maintain a timestamp to provide freshness. The receiver can compare the timestamp to detect the replaying messages easily.

5.2. Message Tampering/Fabrication/Alteration Attack

In our protocol, no matter in which scenario, the messages are hashed using the subgroup key GK or encrypted using the group key r . Therefore, no one can fabricate or modify the messages of vehicles, GLs and the PKG. Also, the subgroup key is updated frequently. Therefore, an intruder could not find the valid subgroup key in a feasible amount of time to communicate with the group members.

5.3. Backward Secrecy

Backward secrecy is the measure of preventing a new vehicle from accessing the previous communication before joining the group. Using key updating process can ensure the backward secrecy. We assume that a vehicle V_j wants to join the subgroup G_i . V_j sends the request message to GL_i which verify its identity. Then GL_i carries out key updating process. It is easy to see that V_j could not obtain the previous subgroup key. When the updated subgroup key is sent to old group members by GL_i , in order to get the old subgroup key, V_j needs to obtain any one of member's pairwise keys. However, the pairwise key is built with K_{GL_i} and the member's three secret keys d_1, d_2 and d_3 . It is impossible to eavesdrop the secret keys, which are not exposed in communication channels. Also, it is impossible to compute the pairwise key due to the hardness of the ECDLP problem. Consequently, the adversary cannot access the communication before joining a subgroup, which means the proposed protocol supports the backward secrecy.

5.4. Forward Secrecy

Forward secrecy is the measure of preventing a vehicle from accessing the current communication after leaving the subgroup. Once a vehicle V_j is revoked from the subgroup G_i network, the associated subgroup key is updated and the current subgroup key is invalidated. In the proposed algorithm, it is infeasible for an adversary to compute the current subgroup key after leaving, because there is no personal keying value $y_j \times m_j$ in GKS_i' , the result that V_j module the new GKS_i' with its secret pairwise key K_{GL-j} is 0.

If one adversary wants to get the new subgroup key, it needs to obtain the value of $\sum_{x=1}^n y_x \times m_x$, where x is the maximum limit of subgroup key value. Note that the value of $\sum_{x=1}^n y_x \times m_x$ is computed by each member's pairwise key, which is not transmitted through the communication channel. Therefore, it is impossible for an adversary to compute the value of $\sum_{x=1}^n y_x \times m_x$. Also it is difficult to compute and eavesdrop the user's pairwise key, which is mentioned above. As a result, the adversary cannot access the current information after leaving a subgroup, which means the proposed protocol supports the forward secrecy.

5.5. Insider Attacks

In our protocol, each message sent by one vehicle incorporates its secret value d_2 and then it is encrypted by the PKG's public key. We assume that if the vehicle V_k is compromised, it tries to forge a vicious message using other's identity. The only way for V_k is to tamper other vehicle's message such as V_j 's in the same subgroup. After eavesdropping the information $\{E_{P_{PKG}}(d_{2j} \parallel P_j \parallel PID_{ji} \parallel M_{public} \parallel t), h_{GK_i}(M_{public} \parallel t), M_{public}, t\}$ sent by V_j , V_k can forge $h_{GK_i}(M_{public}' \parallel t)$ using its shared subgroup key GK_i . Unfortunately, It could not obtain the secret key d_{2j} of V_j because of failing to decrypt the messages using PKG's private key. Moreover, V_k could not change $h_{GK_i}(M_{public} \parallel t)$ to $h_{GK_i}(M_{public}' \parallel t)$, which is also incorporated in $E_{P_{PKG}}(d_{2j} \parallel P_j \parallel PID_{ji} \parallel M_{public} \parallel t)$. Therefore, the proposed protocol can withstand the insider attack.

5.6. Traceability

The proposed protocol provides traceability. If an adversary compromises one vehicle and sends vicious messages to VCs, the compromised ID will be detected by the PKG quickly. When a subgroup leader GL_i considers the message as vicious one, it transmits to the PKG, which acts as the only authorized entity, can perform the tracing procedure and extract the real identity from the information $\{E_{P_{PKG}}(d_{2j} \parallel P_{1j} \parallel PID_{ji} \parallel M_{public} \parallel t), h_{GK_i}(M_{public} \parallel t), M_{public}, t\}$ via decrypting $E_{P_{PKG}}(d_{2j} \parallel P_{1j} \parallel PID_{ji} \parallel M_{public} \parallel t)$. After obtaining d_{2j} and PID_{ji} , the PKG extracts the real identity ID by finding the appropriate $E_{P_{PKG}}(ID)$ to satisfy the relationship $d_2 = (k + E_{P_{PKG}}(ID)) + h_2(P_j, PID_{ji}, T) \times s \bmod q$, in which $E_{P_{PKG}}(ID)$, T and k are stored in its repository. Then the PKG decrypt the $E_{P_{PKG}}(ID)$ and obtain the real identity. Therefore, the proposed method can support the traceability.

5.7. Conditional Privacy

The real identity of a vehicle should be kept anonymous from being revealed by other vehicles and adversaries. When one vehicle shares the information $\{E_{P_{PKG}}(d_{2j} \parallel P_{1j} \parallel PID_{ji} \parallel M_{public} \parallel t), h_{GK_i}(M_{public} \parallel t), M_{public}, t\}$ in the subgroup, due to the anonymous character of the HAMC using subgroup key GK_i , other vehicles can authenticate the vehicle without learning anything about its identity. In the third scenario, if the treacherous vehicle V_k in the same subgroup wants to use the V_j 's identity to forge the malicious messages via eavesdropping $\{E_{P_{PKG}}(d_{2j} \parallel P_{1j} \parallel PID_{ji} \parallel M_{private} \parallel t), h_{GK_i}(PID_{ji} \parallel t), PID_{ji}, t\}$, it will fail to make the right $E_{P_{PKG}}(d_{2j} \parallel P_{1j} \parallel PID_{>ji} \parallel M_{private} \parallel t)$ because of lacking relative secret key d_{2j} of PID_{ji} .

Therefore, the identity of a message sender can be protected from other vehicles, and the CC can trace the real identity of a message sender easily, which is mentioned in Section 5.6. Therefore, the proposed protocol supports the conditional privacy in AVCs.

5.8. Resistance to Black-Hole Attacks

Black-hole attacks are one of the emerging security threats in networks. They occur when a node blocks or drops the packets it receives instead of forwarding them towards the receiving node, which leads to performance degradation in network efficiency and excessive energy consumption [42]. The proposed two-layer architecture can be in favor of the B-H GL nodes detection due to the B-H attack always happens to the intermediate node, the GL node. When transmitting public traffic messages to the CC by GL nodes, in case of receiving time being longer than predefined time interval, a notification of B-H nodes will be sent to the vehicle nodes and the key updating will be performed by the CC. When transmitting subgroup sharing information to the CC by GL nodes, if the B-H GL node does not feedback messages in time to its members, they can notify the CC about the B-H GL node encrypted by the secret key d_2 .

6. Comparison with Related Work

In this subsection, due to the variability of each scheme, we choose some important steps of the protocol to compare a performance with other relative schemes. Because they are time-consuming portions in each of the schemes, we focus on the computation overhead and transmission overhead.

We implement the proposed protocol using a Lenovo computer (Beijing, China) running the VMware Ubuntu12.03 operating system. The machine is equipped with an Intel I7 dual-core processor, a 2.60 GHz clock frequency and 1 gigabytes of memory. For the public key authentication and encryption scheme based on ECC, we use a secp256r1 elliptic curve with the additive group G generated by a point p with the order q , in which p and q are two 256-bit prime numbers to achieve the security level of 128 bits.

6.1. Communication Overhead

6.1.1. Information Transmitting Process

In our proposed model, there is no infrastructure. According to the communication range that is typically 250 m~1000 m [20], vehicles in one group can keep in communication for longer time and remain a more stable group structure because of the relative moving speed and locations. In this structure, we care more about the computation overhead of the group leader. As a group leader, it has much work to do such as forwarding messages for members to the CC, classifying different information received from its members, computing pairwise keys with its members, verifying group members, updating group keys when other vehicle joins into the communication range or any group member leaves the group.

We consider two kinds of information transmission (see 1 and 3 of Section 4.4.6) in our proposed protocol. In the two processes, the computation time at the GL side is important, because we hope that the GL had better not to take longer time to process messages. We consider the computation time of GLs and vehicles respectively, and compare it with the relative methods in [30,31].

For convenience, we define some notations as follows. Let T_{HMAC} denote the time to execute keyed hash message authentication code, T_{Ep} be the time of encryption by public key, T_h be the time of general hash function operation. Then T_{Ek} and T_{Dk} denote the time of encryption and decryption (Advanced Encryption Standard, AES) by symmetric key, respectively. The execution time of aforementioned operations is listed in Table 4.

Table 4. Execution time of different operations.

Operation	Execution Time (Microsecond)
T_{HMAC}	2.6
T_{Ep}	4.5×10^{-6}
T_h	2.3
T_{Ek}	5
T_{Dk}	12
T_s	0.01

The protocol proposed in [31] ensures the security of messages sent from the TA. However, it is a one-way transmission mode because there is no description about how vehicles send messages like service requests to the TA. The proposed scheme in [30] utilizes the RSU to share the traffic information, and it is the group-based mechanism where the RSU acts as the group leader. When the messages are sent from vehicles to the RSU, it will be routed through intermediate nodes, which will forward the messages attaching its signature to the objective RSU. We will compare the computation time of the group leader and the vehicle node when the messages are sent from vehicles to the CC (see Figure 5).

Figure 5. Comparison of computation time when send messages from a vehicle to the center cloud: (a) The computation time at the group leader side; (b) The computation time at the vehicle side.

The relationship between the computation time and the number of messages varying from 10 to 30 is shown in Figure 5. In the Figure 5a, it is obvious that the computation time at the group leader side under the proposed protocol is much less than the one in Lim's scheme [30]. In Figure 5b, the computation time at the vehicle side, which is 0.0076 ms for one message, is 0.0003 ms longer than the one in Lim's scheme. However, the difference is slight, which can be negligible, especially for cars with high technology today.

Figure 6 shows the comparison of the schemes in [30,31] and our protocol about the computation time at the group leader side and the vehicle side when messages sent from the CC to one vehicle.

In Figure 6a, the group leader or intermediate node in our proposed protocol and [30] only transmits messages to a vehicle and does not take time to execute them. It can reduce group leaders' burdens more than the method in [31]. In Figure 6b, the proposed protocol and the scheme in [31] require less computation time to decrypt and verify messages than the method in [30]. Therefore, our protocol can get better effect on the computation time in comparison with the two current schemes.

Figure 6. Comparison of computation time when send messages from the center cloud to a vehicle: (a) The computation time at the group leader side; (b) The computation time at the vehicle side.

In our proposed protocol, we also consider about an important scenario. In cities, vehicles always line up when waiting at traffic lights or being stuck in traffic jams. The distance between two cars are closer and the group members will be increased within the communication range. Therefore, vehicles will send repetitive messages about the same phenomena. In the proposed protocol, it is convenient for group leaders to filtrate the same messages and verify them quickly. It can reduce the heavy burdens of the center cloud to process so many repetitive messages. Figure 7 shows that when the group leader receives repetitive messages, how the computation time changed. We compare it with the existing scheme in [30].

Figure 7. The relationship between repetitive messages and computation time.

In Figure 7, it is obvious to see the advantage of our protocol that when the number of repetitive messages is increased, the computation time of the group leader is decreased.

6.1.2. Key Updating

For key management schemes, key-updating operation happened once the member of the group joins or leaves and it is the most time consuming portion of each scheme. In this subsection,

we benchmark the computation time for updating group keys with two previous similar methods in [31,34]. Unfortunately, the scheme [30] does not provide a group key updating process when one member leaves or joins the group, and it is not secure because the vehicle can access the group messages easily after leaving the group. Also, the vehicle can access the previous messages after joining the group. In the scheme [31], RSUs, act as the group leader, are fixed infrastructure whereas vehicles are mobile. When the vehicle drives from origin to destination, it will pass many RSUs and change the group frequently. The vehicle performs the key updating operation once to change the group. The key updating method in the scheme [34] is more complex. When the member leaves or joins the group, the group leader will perform the key updating operation, which encrypts the new group key and sends it out one by one.

We compare the computation time of group leaders (or intermediate nodes) and vehicle nodes for updating the group key with [34]. Figure 8 shows the relationship between the computation time of the group leader when vehicles join and leave from the subgroup.

Figure 8. Computation time of the group leader during key updating.

In [34], whenever the vehicles join or leave, the group leader will find a new key and encrypt it with different pairwise keys, then send it to the group members, respectively. In our protocol, no matter how many members join or leave, the group leader uses only one subtraction or addition operation to compute the new parameter and transmit relative messages between the CC and subgroup members. In Figure 8, it is obvious that the computation time is less than the scheme in [34] during key updating operation. For subgroup members, in our protocol, one just needs one mod operation to recover the new subgroup key. However, it takes one decrypting operation to obtain the new key in [34]. Therefore, the proposed protocol is a lightweight key management method and is more suitable for vehicular cloud communication.

6.2. Transmission Overhead

In this subsection, we analyze the transmission overhead in our protocol and compare it with other proposed schemes. The transmission overhead includes sending one message from vehicle nodes to group leaders and group leaders to the center cloud. In our experiment, let the length of p , q and M be 256 bits, which is the same as the HMAC. The length of element of G is 512 bits and the timestamp is 32 bits. The length of certificates is 168 bytes according to the IEEE standard. Table 5 describes the results of the transmission overhead in the three scenarios respectively and the comparison between the schemes in [29–31].

Table 5. The comparison of transmission overhead.

Method	Communication Overhead			Result (byte)
Vijayakumar's Scheme	$256 + 256 + 1344 + 256 + 32 + 32 = 2176$ bits = 272 bytes			272
Lim's Scheme	One hop	$256 + 256 + 256 + 32 + 32 = 832$ bits = 104 bytes		104
	N hops	$104 + 121n$ bytes		$104 + 121n$
Yang's Scheme	$512 + 256 + 512 + 512 + 32 + 512 + 512 = 2848$ bits = 356 bytes			356
Our Scheme	Scenario 1	V2G	$256 + 512 + 512 + 256 + 32 + 256 + 256 + 32 = 2112$ bits = 264 bytes	264
		G2C	$256 + 512 + 512 + 256 + 32 + 256 + 32 = 1856$ bits = 232 bytes	232
	Scenario 2	V2G	$256 + 512 + 512 + 256 + 32 + 256 + 32 + 32 = 1888$ bits = 236 bytes	236
		G2C	$256 + 512 + 512 + 256 + 32 + 512 + 32 = 2112$ bits = 264 bytes	264
		G2V	$256 + 256 + 32 + 32 + 32 = 608$ bits = 76 bytes	76
	Scenario 3	V2G	$256 + 512 + 512 + 256 + 32 + 256 + 512 + 32 = 2368$ bits = 296 bytes	296
		G2C	$256 + 512 + 512 + 256 + 32 + 512 + 32 = 2112$ bits = 264 bytes	264

V2G: A vehicle node sends a message to its group leader; G2C: The group leader sends a messages to the center cloud; G2V: The group leader sends a message to the vehicle node.

The average transmission cost of our protocol is 233.1 bytes (including three scenarios) when a vehicle sends one message to others, which is lower than the scheme in [29,31]. In Lim's scheme, if the destination RSU is out of the vehicle communication range, the intermediate vehicle node will forward the message appends its signature based on the onion signature scheme. Therefore, the transmission overhead will increase with the increasing hops.

7. Simulation

We developed a simulator to evaluate the key-management-related computations of messages delivery in the urban scenario. We simulate the proposed scheme in the environment of MATLAB where there is a total number of registered vehicles. We varied the number of vehicles from 30 to 300 moving with 8.3 m/s (30 km/h) to 19.4 m/s (70 km/h) speed on average. The detailed simulation parameter settings are shown in Table 6.

Table 6. Simulation parameter settings.

Parameters	Values
Simulation area	1000 m × 1000 m
Wireless protocol	802.11 p
Channel bit rate	6 Mbs
Numbers of vehicles	(30, 60, 90, 120, 150, 180, 210, 240, 270, 300)
Vehicles speed	30–70km/h
Simulation time	100 s
Radio coverage	250 m

The simulation road is shown in Figure 9, in which the distance between the two roads is 250 m. The initial positions of vehicles are randomly generated and the vehicles are distributed on the road and move to the crossings randomly.

Figure 9. Road configuration for simulation.

Let MD be the total message delay from one vehicle to another vehicle or the center cloud. The message delay can be estimated as the computation time mentioned and the transmission time mentioned in Section 6. In the proposed protocol, there is a difference between the group leader and the node vehicle. They have different tasks and the computation time. Hence, according to the subgroup formation protocol [36], the numbers of group leader and node vehicles are shown in Figure 10 depending on the number of vehicles and the moving speed. Note that each point is running for ten simulations and averaging to get the final result.

Figure 10. The relationship between group leaders and vehicle nodes.

Figure 10 shows the member constitution in each subgroup depending on the increasing number of vehicles. In our simulation, the fewer vehicles are on the road, the more group leaders are selected due to the communication range is 250 m. With the increasing number of vehicles, the ratio of the group leaders and the vehicle nodes is decreased and the repetitive messages in one subgroup are increased. According to the result of the graph, Figure 11 is the simulation result of the average message delay in the three scenarios mentioned in Section 4.4.6.

Figure 11. Average messages delay among different scenarios.

The relationship between MD and the number of vehicles is described in Figure 11, where the number of vehicles varies from 30 to 300 with the average speed from 70 km/h (19.4 m/s) to 30 km/h (8.3 m/s). As shown in Figure 11, the MD in the three scenarios is increased with the increasing number of vehicles. However, the MD is changing a little when the number of vehicles varies from 270 to 300, and it increases by 1 μ s of each vehicle especially in Scenario 2. Also, we can

easily find that the MD growth rate is reducing. The reason is that when the ratio of group leader is decreased, the computation time of authentication and encryption at the group leader side is decreased. Also, the repetitive messages are rising in each subgroup in Scenario 1. Therefore, in the urban scenario, with the increasing number of vehicles, the proposed protocol can undertake the secure communication with the low message delay in the autonomous vehicular clouds.

8. Conclusions

In this paper, we propose an efficient group key management protocol for improving the security of vehicles in AVCs. According to the particularity of AVCs, we apply the CRT and CLPKC to ensure the efficient key management, message authentication and encryption/decryption. In addition, to provide a more stable communication structure for the high mobility of vehicles, we apply a two-layer architecture in AVCs. The vehicles are self-organized within the communication range, hence it can reduce redundant messages and the computation overhead. Furthermore, the proposed protocol divides the messages into four kinds and the way of messages transmission into three kinds with different propose, which can increase the security and efficiency. At last, from the result of the experiment, the computation time at the group leader side and vehicle node side in the different process is much less than that of other well-known schemes. Therefore, our protocol can obtain better trade-offs between security and efficiency and is more suitable for AVCs.

In the future, we will continue our research in other autonomous vehicular cloud scenarios. With the optimization of AVC frameworks, we will study on F2F communication and improve the quality of transmission for V2V and V2I. Autonomous vehicular clouds are one of the examples of the hybrid vehicular networks. We will work on secure communications in the inter-networking scenarios and focus on more secure applications in this hybrid networks.

Author Contributions: C.L. wrote the paper with the assistance of S.J. S.J. designed the experiments and H.W. performed the experiments, D.L. and X.Z. analyzed the data, with assistance of H.L.

Funding: This research received no external funding.

Acknowledgments: All authors, especially the corresponding author Shouwen Ji, would like to thank the anonymous reviewers for their time and invaluable comments and suggestions on this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yu, R.; Zhang, Y.; Gjessing, S.; Xia, W. Toward cloud-based vehicular networks with efficient resource management. *IEEE Netw.* **2013**, *27*, 48–55. [[CrossRef](#)]
2. Kong, Q.; Lu, R.; Zhu, H.; Alamer, A.; Lin, X. A Secure and Privacy-Preserving Incentive Framework for Vehicular Cloud on the Road. In Proceedings of the Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
3. Ahmed, Z.E.; Saeed, R.A.; Mukherjee, A. Challenges and Opportunities in Vehicular Cloud Computing. In *Vehicular Cloud Computing for Traffic Management and System*; Grover, J., Vinod, P., Eds.; IGI Global: Hershey, PA, USA, 2018; pp. 57–74. ISBN 13 9781522539810.
4. Olariu, S.; Eltoweissy, M.; Younis, M. Towards autonomous vehicular clouds. *EAI Endorsed Trans. Mob. Commun. Appl.* **2011**, *11*, e2. [[CrossRef](#)]
5. Alamer, A.; Deng, Y.; Lin, X. Secure and privacy-preserving task announcement in vehicular cloud. In Proceedings of the International Conference on Wireless Communications and Signal Processing, Nanjing, China, 11–13 October 2017; pp. 1–6.
6. Lim, K. Secure and Authenticated Message Dissemination in Vehicular ad hoc Networks and an Incentive-Based Architecture for Vehicular Cloud. Ph.D. Thesis, University of Kentucky, Lexington, KY, USA, 2016.
7. Yan, G.; Wen, D.; Olariu, S.; Weigle, M.C. Security challenges in vehicular cloud computing. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 284–294. [[CrossRef](#)]

8. Rajeshwari, P. A Survey on Security challenges in Vehicular cloud computing. *Int. J. Sci. Res. Educ.* **2016**, *4*, 4848–4853. [[CrossRef](#)]
9. Ahmad, I.; Noor, R.M.; Ali, I.; Qureshi, M.A. The Role of Vehicular Cloud Computing in Road Traffic Management: A Survey. In Proceedings of the International Conference on Future Intelligent Vehicular Technologies, Islamabad, Pakistan, 17–19 October 2017; pp. 123–131.
10. Sookhak, M.; Gani, A.; Buyya, R. A survey on vehicular cloud computing. *J. Netw. Comput. Appl.* **2014**, *40*, 325–344.
11. Wan, J.; Zhang, D.; Zhao, S.; Yang, L. Context-aware vehicular cyber-physical systems with cloud support: Architecture, challenges, and solutions. *Commun. Mag. IEEE* **2014**, *52*, 106–113. [[CrossRef](#)]
12. Chi, W.D.; Ru, L.I.; Fan, P.F. Research on the Checkpoint Server Selection Strategy Based on the Mobile Prediction in Autonomous Vehicular Cloud. In Proceedings of the International Conference on Service Science, Technology and Engineering, Suzhou, China, 14–15 May 2016; pp. 262–267.
13. Hussain, R.; Son, J.; Eun, H.; Kim, S.; Oh, H. Rethinking Vehicular Communications: Merging VANET with cloud computing. In Proceedings of the IEEE International Conference on Cloud Computing Technology and Science, Bristol, UK, 2–5 December 2013; pp. 606–609.
14. Al-Riyami, S.S.; Paterson, K.G. *Certificateless Public Key Cryptography*; Springer: Berlin, Germany, 2003; pp. 452–473.
15. Diffie, W.; Hellman, M.E. New directions in cryptography. *IEEE Trans. Inf. Theor.* **1976**, *22*, 644–654. [[CrossRef](#)]
16. Aloqaily, M.; Kantarci, B.; Mouftah, H.T. A Generalized Framework for Quality of Experience (QoE)-Based Provisioning in a Vehicular Cloud. In Proceedings of the IEEE International Conference on Ubiquitous Wireless Broadband, Montreal, QC, Canada, 4–7 October 2015; pp. 1–5.
17. Aloqaily, M.; Ridhawi, I.A.; Kantarci, B.; Mouftah, H.T. Vehicle as a resource for continuous service availability in smart cities. In Proceedings of the IEEE the 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, Montreal, QC, Canada, 8–13 October 2017; pp. 1–6.
18. Sánchez-García, J.; García-Campos, J.M.; Gutiérrez, D.; Toral, S.; Barrero, F. On-siteDriverID: A Secure Authentication Scheme based on Spanish eID Cards for Vehicular Ad Hoc Networks. *Future Gener. Comput. Syst.* **2016**, *64*, 50–60. [[CrossRef](#)]
19. Wang, L.; Liu, G.; Sun, L. A Secure and Privacy-Preserving Navigation Scheme Using Spatial Crowdsourcing in Fog-Based VANETs. *Sensors* **2017**, *17*, 668. [[CrossRef](#)] [[PubMed](#)]
20. Otoum, S.; Kantarci, B.; Mouftah, H. Adaptively Supervised and Intrusion-Aware Data Aggregation for Wireless Sensor Clusters in Critical Infrastructures. In Proceedings of the IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018.
21. Lin, X.; Sun, X.; Wang, X.; Zhang, C.; Ho, P.H.; Shen, X. TSVC: Timed efficient and secure vehicular communications with privacy preserving. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 4987–4998. [[CrossRef](#)]
22. Cao, X.; Kou, W.; Du, X. A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges. *Inf. Sci.* **2010**, *180*, 2895–2903. [[CrossRef](#)]
23. Chim, T.W.; Yiu, S.M.; Hui, L.C.K.; Li, V.O.K. SPECS: Secure and privacy enhancing communications schemes for VANETs. *Ad Hoc Netw.* **2011**, *9*, 189–203. [[CrossRef](#)]
24. Horng, S.J.; Tzeng, S.F.; Pan, Y.; Fan, P.; Wang, X.; Li, T.; Khan, M.K. b-SPECS+: Batch verification for secure pseudonymous authentication in VANET. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1860–1875. [[CrossRef](#)]
25. Tsai, J.L.; Lo, N.W. A privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE Syst. J.* **2015**, *9*, 805–815. [[CrossRef](#)]
26. Park, Y.; Sur, C.; Noh, S.W.; Rhee, K.H. Self-Controllable Secure Location Sharing for Trajectory-Based Message Delivery on Cloud-Assisted VANETs. *Sensors* **2018**, *18*, 2112. [[CrossRef](#)] [[PubMed](#)]
27. He, D.; Zeadally, S.; Xu, B.; Huang, X. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2681–2691. [[CrossRef](#)]
28. Lo, N.-W.; Tsai, J.-L. An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1319–1328. [[CrossRef](#)]
29. Ming, Y.; Shen, X. PCPA: A Practical Certificateless Conditional Privacy Preserving Authentication Scheme for Vehicular Ad Hoc Networks. *Sensors* **2018**, *18*, 1573. [[CrossRef](#)] [[PubMed](#)]

30. Lim, K.; Manivannan, D. An efficient protocol for authenticated and secure message delivery in vehicular ad hoc networks. *Veh. Commun.* **2016**, *4*, 30–37. [\[CrossRef\]](#)
31. Vijayakumar, P.; Azees, M.; Kannan, A.; Deborah, L.J. Dual Authentication and Key Management Techniques for Secure Data Transmission in Vehicular Ad Hoc Networks. *IEEE Trans. Intell. Trans. Syst.* **2016**, *17*, 1015–1028. [\[CrossRef\]](#)
32. Zheng, X.; Huang, C.T.; Matthews, M. In Chinese remainder theorem based group key management. In Proceedings of the Southeast Regional Conference, Winston-Salem, NC, USA, 23–23 March 2007; pp. 266–271.
33. Zhou, J.; Ou, Y. Key Tree and Chinese Remainder Theorem Based Group Key Distribution Scheme. *J. Chin. Inst. Eng.* **2009**, *32*, 967–974. [\[CrossRef\]](#)
34. Seo, S.H.; Won, J.; Sultana, S.; Bertino, E. Effective Key Management in Dynamic Wireless Sensor Networks. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 371–383.
35. Lim, K.; Abumuhfouz, I.M.; Manivannan, D. Secure Incentive-Based Architecture for Vehicular Cloud. In Proceedings of the International Conference Ad-hoc, Mobile, and Wireless Networks (ADHOC-NOW 2015), Athens, Greece, 29 June–1 July 2015; pp. 361–374.
36. Sahoo, R.R.; Panda, R.; Behera, D.K.; Naskar, M.K. A trust based clustering with Ant Colony Routing in VANET. In Proceedings of the 3rd International Conference on Computing Communication & Networking Technologies, Maui, HI, USA, 30 January–2 February 2012; pp. 1–8.
37. Tran, T.X.; Hajisami, A.; Pandey, P.; Pompili, D. Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges. *IEEE Commun. Mag.* **2017**, *55*, 54–61. [\[CrossRef\]](#)
38. Ridhawi, I.A.; Aloqaily, M.; Kotb, Y.; Ridhawi, Y.A.; Jararweh, Y. A collaborative mobile edge computing and user solution for service composition in 5G systems. *Trans. Emerg. Telecommun. Technol.* **2018**, *1*, 3446–3465. [\[CrossRef\]](#)
39. Ridhawi, I.A.; Samaan, N.; Karmouch, A. Simulator-Assisted Joint Service-Level-Agreement and Vertical-Handover Adaptation for Profit Maximization. In Proceedings of the 12th International Symposium on Applications and the Internet, Izmir, Turkey, 16–20 July 2012; pp. 74–82.
40. Li, C.; Zhang, X.; Wang, H.; Li, D. An Enhanced Secure Identity-Based Certificateless Public Key Authentication Scheme for Vehicular Sensor Networks. *Sensors* **2018**, *18*, 194. [\[CrossRef\]](#) [\[PubMed\]](#)
41. Ding, C.; Pei, D.; Salomaa, A. Chinese Remainder Theorem. *World Sci.* **2005**, *70*, 224.
42. Otoum, S.; Kantarci, B.; Mouftah, H.T. In Hierarchical trust-based black-hole detection in WSN-based smart grid monitoring. In Proceedings of the IEEE International Conference on Communications, Paris, France, 21–25 May 2017; pp. 1–6.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).