Edge Computing in VANETs-
An Efficient and Privacy-Preserving Cooperative Downloading Scheme

Jie Cui, Lu Wei, Hong Zhong, Jing Zhang, Yan Xu, Lu Liu

Abstract

With the advancements in social media and rising demand for real traffic information, the data shared in vehicular ad hoc networks (VANETs) indicate that the size and amount of requested data will continue increasing. Vehicles in the same area often have similar data downloading requests. If we ignore the common requests, the resource allocation efficiency of the VANET system will be quite low. Motivated by this fact, we propose an efficient and privacy-preserving data downloading scheme for VANETs, based on the edge computing concept. In the proposed scheme, a roadside unit (RSU) can find the popular data by analyzing the encrypted requests sent from nearby vehicles without having to sacrifice the privacy of their download requests. Further, the RSU caches the popular data in nearby qualified vehicles called edge computing vehicles (ECVs). If a vehicle wishes to download the popular data, it can download it directly from the nearby ECVs. This method increases the downloading efficiency of the system. The security analysis results show that the proposed scheme can resist multiple security attacks. The performance analysis results demonstrate that our scheme has reasonable computation and communication overhead. Finally, the OMNeT++ simulation results indicate that our scheme has good network performance.

Index Terms

VANETs, data downloading, security, privacy-preserving, edge computing, fuzzy logic.

Jie Cui, Lu Wei, Hong Zhong, Jing Zhang, and Yan Xu are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, the School of Computer Science and Technology, Anhui University, Hefei 230601, China, the Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China, and the Institute of Physical Science and Information Technology, Anhui University, Hefei 230601, China. (e-mail: cuijie@mail.ustc.edu.cn; dreamer_weilu@163.com; zhongh@ahu.edu.cn; root_zj@163.com; xuyan@ahu.edu.cn).

Lu Liu is with the School of Informatics, University of Leicester, LE1 7RH, UK (email: l.liu@leicester.ac.uk).

I. Introduction

With an increase in the number of vehicles and the rising popularity of on-board applications, vehicular ad hoc networks (VANETs) have attracted significant attention in the recent years. As an important application of the Internet of Things (IoT), hardware devices and embedded systems installed in vehicles enable communication among entities in VANETs; thus, significantly improving the safety of drivers. Two of the most common communications of VANETs are vehicle-to-vehicle (V2V) communications and vehicle-to-infrastructure (V2I) communications [1] [2].

In V2V communications, beacons or otherwise called basic safety messages (BSMs) containing the velocity, steering-wheel angle, brake system status, etc. [3] are exchanged periodically among the vehicles. V2V communications are insufficient in ensuring the drivers safety because each vehicle can only obtain information from the nearby vehicles. As a supplement to V2V communications, V2I communications allow vehicles to request traffic-related and entertainment-related information from the nearby infrastructures, which can improve the traffic safety and driving experience.

Data downloading is a promising and practical application in VANETs. Further researches on automatic driving have revealed that the demands for data downloading have been on a rise. This is so because real-time traffic status awareness is based on the availability of image or video recognition [4]. For V2I communications, cellular networks and the 802.11p wireless access are the two most common types of communications. Vehicles can obtain traffic-related and entertainment-related content directly from the service providers by using a cellular network, or indirectly from roadside infrastructures by using IEEE 802.11p protocol. Due to the high speed of the vehicles, downloading bulk data, such as videos or high-quality images, using a cellular network can be expensive. In addition, when mobile data demands are larger than usual, a cellular network can face network jam problems.

An efficient method to solve the cellular network problem is enabling the vehicles to retrieve content from a roadside unit (RSU), instead of obtaining it directly from the content service provider (CSP) using a cellular network. The links between the RSUs and vehicles have a relatively high bandwidth and low latency. The data downloading ability provided by RSUs to the nearby vehicles is also known as drive-thru Internet [5], which has received considerable attention in the past decade. However, this technique has its own innate defects. Firstly, the number of RSUs deployed along the road is limited. Secondly, excessive data access demands might exceed the network loads of RSUs.

To address the limitations of drive-thru Internet, a cooperative downloading method can be used. As known, vehicles have large velocities in a real scenario, which leads to momentary link connections with the RSU. For large size data, there is a high probability that vehicles would fail to download the data completely. Besides, the vehicles that are on the edge of the communication coverage of the RSU have a weak communication link, which means that it is difficult for these vehicles to directly download data from the RSU. With the use of a cooperative downloading method, vehicles that have good communication links with the RSU can help the nearby vehicles to download data. Further, this technique can reduce the medium access control (MAC) layer collisions, and enhance the network reliability and transmission throughput of VANETs [6]–[8].

Security and privacy are two critical issues in the cooperative downloading research area of VANETs. The data downloaded from vehicles are mainly traffic-related or entertainment-related. For traffic-related data, any modification or injection may negatively affect the traffic safety, and the situation is more serious in case of autonomous driving as the modification of the sensor or camera data can alter the driving strategy. For entertainment-related data, the downloading requests are often related to identity privacies. Therefore, it is necessary to consider the security and privacy factors in cooperative data downloading of VANETs [9] [10].

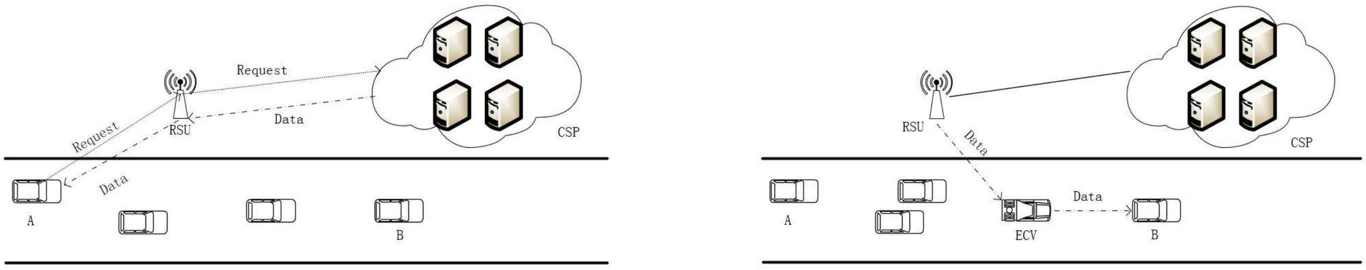
Several representative schemes have been proposed to solve the security and privacy issues of data downloading in VANETs. Hao et al. [9] were the first to consider these issues in the data downloading area of VANETs. In their proposed scheme, they used elliptic curve, symmetric key encryption, and a hash function to secure data downloading between the vehicles and RSU, thus, guaranteeing the privacy of the vehicles requests and authentication of the downloaded data. However, their proposed scheme did not support cooperative downloading and failed to satisfy several security requirements such as conditional privacy-preserving and resistance to the man-in-the-middle attack. Hao et al. [11] further developed a secure cooperative data downloading framework. They used bilinear pairing-based broadcast encryption and symmetric key encryption algorithms to ensure secure data downloading, and proposed a location-based data sharing protocol where relay and downloading vehicles could collaborate to download data. However, in terms of security and privacy, their proposed scheme could only consider the confidentiality of the downloaded data and did not consider its authentication, and, thus, failed to satisfy some necessary security requirements of VANETs. Lai et al. [12] proposed a secure incentive scheme for reliable cooperative downloading. In this scheme, few proxy vehicles were elected to assist the nearby vehicles to cooperatively download data fairly and reliably. Further, a Camenisch-Lysyanskaya (CL) signature-based incentive mechanism was designed to stimulate cooperation and reduce the authentication overhead. However, as these schemes were constructed on bilinear pairing, the operation costs involved were relatively large.

The existing representative secure cooperative data downloading schemes [11] [12] have the following limitations. Firstly, they are not lightweight in terms of the cryptography methods being used. Secondly, they do not consider the detailed election strategy of the relay vehicles that is responsible for cooperative downloading data. It needs to be noted that not all of the vehicles are suitable to act as relay vehicles. Besides, the data downloading processes in their respective schemes require RSU being online all the time. Thirdly, the most important limitation is that a critical phenomenon has not been considered in these schemes, i.e., as mentioned by Magaia et al. [13], vehicles in the same area will often have the same requests for some popular data, which means that these popular data could be downloaded with a high frequency. For example, vehicles in the same area often care about local traffic information, such as future weather and local news. If we neglect the data request feature, the downloading efficiency of the entire system will be relatively low.

To solve the limitations of the existing schemes, we propose a secure and privacy-preserving cooperative downloading scheme that considers high-frequency requested data and is constructed using lightweight cryptography methods. The primary process of the proposed scheme can be divided into two stages, namely a non-accelerated stage and an accelerated stage. In the non-accelerated stage, the CSP periodically broadcasts a content list that mentions the provided data. If a vehicle wants to download the data contained in the list, it sends encrypted requests to the CSP and local RSU, respectively, and downloads the data from the CSP through the RSU. After collecting a certain number of encrypted requests, the system enters the accelerated stage. In this stage, the RSU analyzes the encrypted requests sent from the nearby vehicles to obtain the request frequency information. It, then, selects appropriate qualified vehicles called edge computing vehicles (ECVs) to buffer the high-frequency requested data. If a vehicle is near ECVs, it can directly download the popular data from the nearby vehicles that store them. Hence, the downloading process for popular data gets accelerated. For popular data provided by a CSP, ECVs act not just consumers, which download data from the CSP, but also data providers, which provide a partial downloading service for the nearby vehicles. The essence of the idea lies in the use of the edge computing concept.

A. Contribution

The main contributions of this study are as follows:



(a) non-accelerated stage

(b) accelerated stage

Fig. 1. Data downloading phases

1) Firstly, we consider the high-frequency requested (popular) data phenomenon in the secure cooperative downloading scenario of VANETs, i.e., vehicles in the same area often have the same requests for some popular data. Further, we propose an edge computing-based secure and privacy-preserving cooperative downloading scheme to accelerate the data downloading process for popular data.
2) The proposed scheme is constructed using lightweight cryptography methods, such as elliptic-curve cryptography, the TESLA broadcast authentication, and additive homomorphic encryption algorithms, instead of computationally expensive bilinear pairing-based cryptosystems.
3) A fuzzy-logic-based election strategy is designed to select qualified vehicles (ECVs) for collaborative downloading where popular data can be directly downloaded from ECVs. Compared to the existing schemes that rely on vehicles in the transmission path and the RSU simultaneously for collaborative downloading, the proposed scheme is identified to be more efficient.
4) Security analysis shows that the proposed scheme can satisfy all the necessary security requirements of VANETs. Further, the results of OMNeT++ simulation experiments demonstrate that the proposed scheme shows network performance benefits over the existing representative schemes.

B. Organization

In section II, we introduce the related work. In section III, the background of the proposed scheme is demonstrated. The main process of the proposed scheme can be divided into two parts, the non-accelerated and accelerated stages, as shown in section IV and V, respectively. In section VI, we prove and analyze the security of the proposed scheme. In section VII, we analyze the performances of the proposed scheme. Finally, the conclusion of the proposed scheme is provided in section VIII.

II. Related Work

A. Data Sharing/Downloading Research

Zhang et al. [14] proposed a novel protocol called vehicular cooperative media access control, which used the cooperative communication concept to solve the entertainment information and television communication of VANETs. In their paper, a part of trusted users act as relay vehicles to forward packets sent from nearby vehicles. And the total system throughput can be greatly enhanced than former schemes. Saad et al. [6] first proposed a novel cooperative strategy among RSUs in VANETs while former works focused on non-cooperative methods for vehicle-to-RSU communication. They modeled the issue as a coalition formation game and proposed a solution for coalitions among RSUs based on game theory. Hao et al. [11] proposed a secure cooperative data downloading framework for paid services in VANETs. In their work, vehicles can download data from RSU when they are within the coverage of RSU and share the downloaded data to the vehicles which are out of coverage of RSU. They designed an application layer data sharing protocol which can avoid collisions in MAC layer. Besides, they first addressed the security and privacy issues in data sharing problem of VANETs. Lai et al. [12] proposed a secure incentive scheme to achieve fair and reliable cooperative downloading for VANETs. In their scheme, the virtual check method was used to achieve incentive mechanism which can motivate vehicles to help each other download large files. Besides, the aggregating CL-signature was used to ensure the security of their scheme.

B. Security and Privacy Research

Since messages exchanged by vehicles often involve the safety of drivers, messages need to be authenticated before retrieving the inside content. Raya et al. [15] proposed a message authentication scheme based on public key infrastructure which can guarantee that messages exchanged by vehicles are authenticated. However, their scheme used digital certificates that could bring high computational and storage overhead. Besides, batch authentication was not supported in their scheme that would limit the performance of message authentication. Zhang et al. [16] proposed a RSU-aided message authentication scheme. In their scheme, the authentication tasks of messages exchanged by vehicles were processed by RSU. And batch authentication was supported that could speed up the efficiency of message authentication. The computing capacities of vehicles are limited, which means that vehicles are inadequate to authenticate messages within the specified time in high-density environment. To solve the issue, Chim et al. [17] proposed a novel message authentication scheme based on bloom filter. The message authentication task of messages exchanged by vehicles are processed by RSUs, and vehicles can receive the message authentication results from bloom filters sent by RSUs. The novel method enhanced the efficiency of the entire VANETs system because the redundant authentication is reduced greatly.

Traditional schemes are often based on bilinear pairing that is costly in terms of computational overhead. He et al. [18] proposed an elliptic-curve-cryptography (ECC)-based scheme instead of using bilinear pairings. The computation overhead is significantly reduced by discarding the bilinear paring operation. After the emergence of the scheme proposed by He et al. [18], many improved ECC-based schemes have been proposed. Cui et al. [19] proposed a message authentication scheme based on Cuckoo filters [20] and binary searching method in which the message authentication success rate in the batch authentication phase is much higher than traditional schemes. Zhong et al. [21] proposed a conditional privacy-preserving message authentication scheme based on the registration list. The communication overhead was significantly reduced compared with traditional related schemes, because more lightweight cryptography methods were used. Dua et al. [22] proposed a secure message authentication key exchange scheme based on ECC technique. In their scheme, CH (Cluster Head) vehicles are responsible for messages authentication of vehicles in its proximity that can decrease the computation burden of TA. Although ECC based schemes have the computational advantage over bilinear pairing based schemes, the scheme still have one obvious weakness. As known, signing and verifying one data packet both have relatively high overhead which means that ECC based schemes are not robustness to distribute-denial-of-service (DDoS) attacks and packet loss attacks especially in broadcast application scenario [23]. For example, malicious attackers can deliberately fabricate a lot of invalid signatures which can occupy many CPU resources of vehicles and disturb normal communication. Studer et al. [24] proposed a new hybrid authentication scheme based on ECDSA signing algorithm [25] and TESLA [23]. Their scheme can provide fast authentication and non-repudiation. Chen et al. [26] proposed an efficient broadcast authentication scheme that can defend both computation-based DoS attacks and memory-based DoS attacks. Besides, vehicles can achieve instant verification by designing a mechanism to make vehicles have the ability to predict future beacons in advance.

C. Edge Computing Research

Traditional cloud computing technology is not qualified for the scenario where a large quantity of data are produced. Shi et al. [27] introduced the edge computing concept to Internet of Things (IoT) field. Edge computing or fog computing is gaining popularity and being increasingly deployed in various latency-sensitive application domains including industrial IoT [28]. The edge of the IoT acts as not only the consumer but also the producer, i.e., some traditional cloud services can be transferred to the edge node of network, which can achieve some good effects like better offloading, lower latency and so on. Zhang et al. [29] proposed a hierarchical cloud-based vehicular edge computing framework for offloading service. They used Stackelberg game model to optimize the resource allocation between vehicle edge computing services. Huang et al. [30] proposed a control scheme for offloading vehicular communication traffic. They first proposed an architecture based on Software Define Network (SDN) inside Mobile Edge Computing (MEC) concept. Feng et al. [31] proposed an autonomous vehicular edge framework that aims to manage resources on vehicles efficiently. They designed a scheduling algorithm based on ant colony optimization to solve the resource allocation problem of computing loads of vehicles.

III. Background

A. Network Model

As shown in Fig. 2, the VANETs system model used in our paper mainly contains four kinds of entities: trust authority (TA), roadside unit (RSU), the vehicle equipped with On-Board Unit (OBU), and content service provider (CSP).

- **TA**: TA acts as the registration center of RSUs and OBUs and is trusted by all entities in VANETs. The main responsibility of TA is to issue key material to RSUs and OBUs. The connections between TA and RSU, RSU and OBU are wired connection and wireless connection respectively. The secure communication between TA and RSU is guaranteed by a secure transmission protocol. In addition, to provide services for vehicles, TA also supervises the behavior of vehicles, i.e., it has the ability to reveal the real identity from messages sent from a vehicle. In order to avoid single point failure and improve the system reliability, redundant TAs are usually set up.
- **RSU**: RSU is deployed on the roadside, acts as the bridge between TA and vehicles, and mainly responsible for providing network access service to nearby vehicles. On the one hand, the traffic-related information and entertainment content can be transmitted to vehicles by RSU. On the other hand, driving data collected from sensors on vehicles can be transmitted to DMV by RSU [19]. In our model, RSU is assumed to have enough storage space and computation resources to cache a certain number of data packets.

- **Vehicle**: Vehicles play the most core role that are responsible for periodically broadcasting beacons(or so-called Basic Safety Message) [3] to improve traffic efficiency and driving safety. Each vehicle is equipped with a tamper-proof device (TPD) [32] to store received key material from the TA securely and we assume TPD unhackable. In our model, vehicles are divided into two types: ordinary vehicle (OV) and edge computing vehicle (ECV). A part of ordinary vehicles that have relatively better storage performance, more appropriate velocity and location are elected as ECVs. ECVs cache a part of high-frequency requested data and OVs can directly retrieve data from nearby ECVs if the data have been stored in the buffer memory of ECVs.
- **CSP**: In our scheme, CSP is the generic term of all content service providers, which is responsible for providing traffic-related or entertainment services to vehicles.

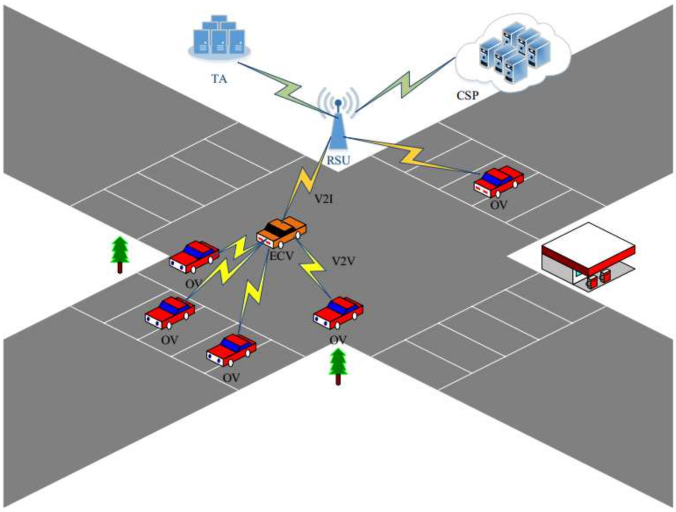


Fig. 2. The model of VANETs.

B. Security Objectives

The proposed scheme can achieve the following security objectives:

1) Message Authentication and Integrity: The receiver authenticates the message to ensure it is complete and has not been tampered with, i.e., the message receiver must determine the legitimacy of the message owner, whether the message is fabricated, and whether the message has been modified before retrieving the content contained in the message.
2) Identity Privacy-Preserving: In order to protect the identity privacy, the vehicle uses the pseudo-identity to send all messages. Any third party except for TA can not calculate the real identity of the vehicle through the messages.

3) Traceability: TA can trace the real identity of the vehicle by analyzing its pseudo identity extracted from its message but any malicious adversary does not have the ability.
4) Replay Attack Resistant: If malicious attackers resend the existing messages, the receiver determines whether the message is expired or not by checking the timestamp attached to the message, and rejects the expired message to avoid a replay attack.
5) DDoS Attack Resistant: The scheme can resist DDoS attack launched by attackers who could send a lot of useless data packets to cause the network congestion.
6) Man-in-the-middle Attack Resistant: The adversary can not forge as a legal entity to modify the messages transmitted between two entities for interrupting the normal communications.

C. Elliptic Curve Cryptosystem

In 1984, Miller applied the elliptic curve to cryptography for the first time [33]. After Kobilitz built the elliptic curve cryptography (ECC) [34] with elliptic curve discrete logarithm problem (ECDLP), ECC began to be widely applied to encryption, protocol, and other safety-related areas. Let F_p be a finite field, which is determined by a prime number p. Let a set of elliptic curve points E over be defined by the equation: $y^2 = x^3 + ax + b \mod p$, where $a, b \in F_p$. Let the point at infinity be O, then O and other points on E make up an additive elliptic curve group $\mathbb{G}$ with the order q and other generator P. The elliptic curve group $\mathbb{G}$ has the three following properties.

- Additive: Let P and Q be two points of group $\mathbb{G}$. If P is not equal to Q, then we can get $R = P+Q$ where R is the intersection of E and the straight line connecting P and Q. If $P = Q$, then $R = P+Q$. If $P = -Q$, then $P + Q = O$.
- Scalar point multiplication: Let $P \in \mathbb{G}$ and $m \in Z_q^*$, the scalar multiplication of E is defined as $m \cdot P = P + P + \cdots + P$.
- Elliptic curve discrete logarithm problem (ECDLP): Given two randomly generated points, it is difficult to calculate $x \in Z_q^*$ in the case of known two points $P, Q = xP \in \mathbb{G}$.
- Elliptic Curve Computational Diffie Hellman Problem (ECCDHP): Given one tuple $\{P, xP \in \mathbb{G}, yP \in \mathbb{G}\}$ where $x, y \in^R Z_q^*$, the advantage for any probabilistic polynomial time (PPT) adversary to calculate $xyP \in \mathbb{G}$ is negligible.

D. TESLA protocol

TESLA [23] is an efficient broadcast authentication protocol that has very low communication and computation overhead compared with traditional signature-based authentication protocol. In TESLA, purely symmetric cryptographic functions are used. The sender first randomly chooses a secret key k_n, calculates $k_{i(i\varepsilon[0,n])} = H^i(k_n)$ in which $H()$ denotes the one-way hash function, and then formulates a hash chain $\{k_0, k_1, ..., k_n\}$ for time interval $\{I_0, I_1, ..., I_n\}$. Moreover, the sender uses a second hash function $H^{'}$ to derive the authentication key $k_i^{'} = H^{'}(k_i)$ which is used to compute the MACs of messages. To authentication the message sent from the sender in I_i, the receiver waits for $d(d \geq 1)$ intervals to get the key k_i for I_i, and check the validity of the message by checking the corresponding MAC using key k_i. Although TESLA is a lightweight authentication scheme that has very low computation overhead, it can't provide non-repudiation. An effective remedy is signed the initial secret key k_0 using a digital signature algorithm.

E. Homomorphic Encryption

Homomorphic encryption is a cryptographic technique based on computational complexity theory of mathematical problems. Processing the encrypted data to obtain an output, decrypting the output, and the result is the same as that obtained by processing the unencrypted data in the same way. In essence, homomorphic encryption refers to such an encryption function, which performs ring addition and multiplication on the plain text and then encrypts it, performs corresponding operations on the ciphertext after encryption, and the result is equivalent. In our paper, we use the Paillier cryptosystem [35] and the basic form is shown as $Dec(Enc(m_1) \cdot Enc(m_2)) = m_1 + m_2$.

F. Fuzzy Logic Control System

A fuzzy control system is a control system based on fuzzy logic mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 1 or 0 (true or false, respectively). Fuzzy logic is widely used in machine control. The term fuzzy refers to the fact that the logic involved can deal with concepts that cannot be expressed as true or false but rather as partially true. Fuzzy logic has the advantage that the solution to the problem can be cast in terms that human operators can understand, so that their experience can be used in the design of the controller. This makes it easier to mechanize tasks that are already successfully performed by humans [36] [37] [38]. A typical fuzzy logic control system mainly contains the following three steps:

- **Fuzzification** Fuzzification refers to the process which converts the crisp input value to the fuzzy value.
- **Fuzzy rules defining** The fuzzy rules are several If-Then statements which input several fuzzy values and output a fuzzy value.
- **Defuzzification** Defuzzification is responsible for choosing an appropriate representative value as a final output which is a crisp value. The most common defuzzification method is the center of gravity (COG) method.

TABLE I
NOTATIONS TABLE

Notations	Definitions
TA,RSU,CSP,V_i	Trusted authority, roadside unit, content service provider, and vehicle respectively
E	An elliptic curve $y^2 = x^3 + ax + b \; mod \; p$
$\mathbb{G}$	An additive elliptic curve generated by base point P with order q
s,P_{pub}	System private key s and public key $P_{pub} = sP$ respectively
x,PK_{RSU}	Private key y and public key PK_{RSU} of RSU respectively
y,PK_{CSP}	Private key y and public key PK_{CSP} of CSP respectively
H_1,H_2,H_3,H_4,H,$H^{'}$	Secure one-way collision-resistant hash functions
lcm, gcd	Least common multiple and greatest common divisor respectively
$MAC_k(\cdot)$	Message authentication code
PID_i,RID_i,PWD_i	The pseudo identity, real identity, and password of vehicle V_i respectively
$\bigoplus$,$\|$	Bitwise xor and concatenation operations respectively
T_i	Message sending timestamp

IV. Non-accelerated Stage of Our Proposed Scheme

The non-accelerate stage of our proposed scheme can be divided into several steps. Firstly, vehicles and RSUs get initialized by the TA, i.e., they receive necessary security parameters and key materials from TA, which are the bases for future secure communications. Secondly, CSP broadcasts the content service list (CSL) that includes the brief introduction of the data it can provide and is secured by TESLA algorithm. Thirdly, vehicles authenticate the legality of CSL, establish the session keys with CSP for encrypting requests to CSP, and send encrypted requests that are secured by a designed elliptic-curve-cryptography-based signature algorithm to RSU and CSP respectively. Finally, vehicles download the content sent from CSP through RSU.

A. System Initialization

At the system initialization phase, both RSUs and vehicles receive some necessary key material and parameters from TA, and vehicles need to choose a pseudo identity. The key material and parameters transmission between TA and RSUs can be processed easily because their connections are special secure wired networks. However, the connections between TA and vehicles are insecure wireless networks which are vulnerable to all kinds of attacks. Hence the initial key material and parameters between TA and vehicles need to be processed on some special occasions such as vehicle inspection under a pre-defined strategy. The detailed steps of system initialization phase are as follows:

1) TA randomly chooses two large primes p and q, a non-singular elliptic curve E which is defined as $y^2 = x^3 + ax + b \, mod \, p$, and a random generator element P that generates the additive elliptic curve group $\mathbb{G}$ with order q.
2) TA randomly chooses the system private key $s \in Z_q^*$ and calculates the corresponding system public key $P_{pub} = sP$.
3) TA randomly chooses RSU's private key $x \in Z_q^*$ and calculates RSU's public key $PK_{RSU} = xP$.
4) TA randomly chooses CSP's private key $y \in Z_q^*$ and calculates CSP's public key $PK_{CSP} = yP$.
5) TA chooses three secure hash functions $H_1 : \mathbb{G} \times \{0,1\}^* \times \{0,1\}^* \to Z_q^*$, $H_2 : \mathbb{G} \times \{0,1\}^* \to Z_q^*$, $H_3 : \mathbb{G} \to Z_q^*$, $H_4 : \{0,1\}^* \times \{0,1\}^* \to Z_q^*$, two secure one-way functions $H, H' : Z_q^* \to Z_q^*$, and MAC function $MAC_k(m) = H_1(k \oplus opad \| H_1(k \oplus ipad \| m))$ where $opad$ and $ipad$ both denote the padding constants.
6) TA assigns unique real identity RID_i and password PWD_i to each vehicle V_i, preloads PWD_i, RID_i, s to the TPD of each vehicle, and sends the private key x, y to RSU and CSP separatively through secure channels.
7) RSU chooses two primes p_1 and p_2 randomly, calculates $n = p_1 \cdot p_2$ and $\lambda = lcm(p_1, p_2)$, chooses a random number $g \in Z_{n^2}^*$, and sends n and g to TA.
8) TA publishes the system public parameters and functions $\{p, q, a, b, n, g, P, \mathbb{G}, P_{pub}, PK_{RSU}, PK_{CSP}, H_1, H_2, H, H', MAC\}$ to all vehicles, RSUs and CSP through public channels.

B. Content List Broadcast

In order to provide the content downloading service to vehicles, CSP needs to broadcast a list that describes what kind of content it can provide. The list in our scheme is called Content Service List

(CSL). The structure of CSL mainly contains three parts: index, description, and length where the index item is used to locate the content, the description item is the short description about the provided content, and the length is the byte size of the content. We give an example of CSL as shown in Table II.

After CSP is ready for CSL, it needs to broadcast CSL to vehicles so that vehicles can download the content of interest. However, CSL is easy to be modified or fabricated if no authentication mechanism piggyback. Since CSL is one kind of broadcast application, broadcast authentication protocol rather than digital signature protocol is more suitable for guaranteeing the authentication of CSL. In our proposed scheme, we use TESLA algorithm [23] to guarantee the authentication of CSL, which can increase the strength to resist DDoS attack. The detailed steps are as follows.

1) *Hash Chain Generation*: Firstly, CSP splits the timeline into a sequence of intervals, which are remarked as $I_0, I_1, ..., I_t$. Secondly, CSP chooses a secret K_t randomly, continues to perform hash operation to produce sequence $\{K_0, K_1, ..., K_t\}$, and gets the chained private key sequence $\{K_0^{'}, K_1^{'}, ..., K_t^{'}\}$ for different time intervals by chained computing $K_{i-1} = H(K_i)(1 \leq i \leq t)$ and $K_i^{'} = H^{'}(K_i^{'})(0 \leq i \leq t)$. The processes are as shown in Fig. 3.
2) *Initial Key Broadcast*: CSP needs to sign the initial key K_0 using ECDSA signing algorithm and its secret key y, and broadcasts message $m_0 = \{T_i, K_0, ECDSA\_Sign_y(K_0^{'} \parallel T_i)\}$ periodically. It is noteworthy that only TESLA algorithm can not provide non-repudiation and a solution is to sign the initial secret key K_0. And periodically broadcast of m_0 is to ensure that the new incoming vehicle can still authenticate the validity of CSL.
3) *CSL Broadcast*: CSP prepares the newest CSL, calculates current time interval I_i, and broadcasts message $m_i = \{T_i, CSL, MAC_{K_i}(CSL|T_i), K_{i-1}\}$.

TABLE II
THE STRUCTURE OF CSL

Index	Description	Length
0	The traffic flow information of Beijing downtown	10KB
1	Piano music	3MB
2	Weather forecast for today	5KB
...	...	...
255	City map of Shanghai	8MB

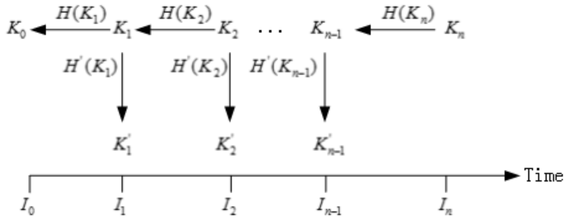


Fig. 3. The hash chain generation process.

C. Content Request from Vehicles

After the vehicle determines the specific content request, it needs to send the request to CSP and RSU. Because the content request involves user privacy, it should be encrypted. Since the content requests sent to RSU are only used to analyze the access frequency and RSU does not have the ability to get the content request of single-vehicle, homomorphic encryption is a good choice to encrypt the request sent to RSU. The detailed steps are as follows.

1) *CSL Authentication*: After receiving CSL from CSP, V_i needs to verify the validity of CSL because CSL may be modified by malicious attackers during transmission process. Firstly, V_i checks the message $m_0 = \{T_i, K_0, ECDSA\_Sign_y(K_0 \parallel T_i)\}$ using ECDSA verifying algorithm and public key PK_{CSP} of CSP. If valid, V_i checks the validity of received m_i. Secondly, V_i verifies the validity of received key K_{i-1} by following the one-way key chain back to K_0. If valid, the received K_{i-1} can be used to check the validity of former message m_{i-1} sent from CSP. In the same way, the validity of m_i can be checked after receiving m_{i+1} in the next time interval.

2) *Key Agreement Between CSP and Vehicles*: Vehicle V_i selects a random number $r_i \in Z_q^*$, and calculates the pseudo identity $PID_i = \{PID_i^1, PID_i^2\}$ where $PID_i^1 = r_i \cdot P$, $PID_i^2 = RID_i \oplus H_3(r_i \cdot P_{pub})$. Next, V_i chooses a random number $z \in Z_p^*$, calculates $Z = zP$ and the signature $\sigma_i = s \cdot H_1(PID_i \parallel T_i) + r_i \cdot H_2(Z \parallel T_i)$, and sends message $\{PID_i, Z, T_i, \sigma_i\}$ to CSP. It is worth noting that PID_i is a one-time pseudo identity, i.e., r_i is regenerated for a new message, to resist chosen message attack (CMA) for the signature. After receiving the message, CSP verifies the message by checking Equation (1). If the condition is true, CSP calculates $K_{CSP} = H(y \cdot Z)$.

$$\sigma_i \cdot P \overset{?}{=} H_1(PID_i \parallel T_i) \cdot P_{pub} + H_2(Z \parallel T_i) \cdot PID_i^1 \tag{1}$$

3) *Encryption of The Request Sent to RSU*: First, V_i converts the index of the content that it wants to download into the binary format. The format of the download requests is expressed as the binary set

$\{S_1(v), S_2(v), ..., S_m(v)\}$, where the last bit of S_i^v denotes whether V_i want to download No.i content and the bit length of S_i^v is equal to the base-2 logarithm of the max number of vehicles that one RSU can provide access services. For example, if V_i wants to download No.1 and No.4 content, the request binary set of V_i is $\{001, 000, 000, 001\}$. Next, V_i selects a random number $x_i \in Z_n^*$ that satisfies $gcd(x_i, n) = 1$ and computes its encrypted request binary set $C_i = g^{S_1(v)\|S_2(v)\|...\|S_m(v)} x_i^n \ mod \ n^2$. Last, V_i signs C_i by computing $\sigma_i = s \cdot H_1(PID_i \parallel T_i) + r_i \cdot H_4(C_i \parallel T_i)$, and sends $\{C_i, T_i, \sigma_i\}$ to RSU.

4) *Encryption of The Request Sent to CSP*: After V_i determines the content it wants to download, it encrypts the index of the content by calculating $C_i^{'} = Enc_{K_{V_i}}(\{S_1(v), S_2(v), ..., S_m(v)\})$, where Enc denotes the AES encryption algorithm and K_i denotes the session key that is computed by $K_i = H(z \cdot PK_{CSP})$. Next, V_i calculates the signature $\sigma_i = s \cdot H_1(PID_i) + r_i \cdot H_4(C_i^{'} \parallel T_i)$, and sends $\{C_i^{'}, T_i, \sigma_i\}$ to CSP.

D. Downloading Process

After CSP receives the encrypted request $C_i^{'}$ from V_i, it firstly verifies the signature by checking the condition $\sigma_i \cdot p \overset{?}{=} H_1(PID_i \parallel T_i) \cdot P_{pub} + H(C_i^{'} \parallel T_i) \cdot PID_i^1$. If valid, CSP will decrypt the encrypted request by computing $Dec_{K_{CSP}}(C_i^{'})$, where Dec denotes the AES decryption algorithm. Next, CSP finds the corresponding content set which is remarked as CS_i, and computes its signature $\sigma_{CSP} = ECDSA\_Sign_y(CS_i)$. Last, CSP sends $\{CS_i, \sigma_{CSP}\}$ to the RSU which is responsible to transmit the content to V_i. After receiving $\{CS_i, \sigma_{CSP}\}$, V_i can check the legality of the data by using ECDSA verification algorithm and PK_{CSP} to check the signature signed by CSP.It is worth noting that the data sent from RSU can be plaintext or ciphertext that could be encrypted using attribute based encryption (ABE) algorithm for access control.

V. Accelerated Stage of Our Proposed Scheme

The accelerated stage of our proposed scheme can be divided into several steps. Firstly, RSU analyzes the downloading frequency of the requested data by resolving the aggregated encrypted requests sent from nearby vehicles, and knows which data are the most popular. Secondly, RSU selects several ECVs from nearby vehicles using a fuzzy-logic-based election method. Finally, ECVs buffer the data that are corresponding to high-frequency requests to nearby ECVs, and the vehicle with requests for these popular data can directly download them from nearby ECVs.

A. *The Analysis of Vehicle Requests by RSU*

After RSU gets enough encrypted requests of nearby vehicles, it can analyze the frequency of recent data downloading. Let k denote the number of received downloading request sets in one period. Firstly, RSU authenticates the legality of the batch of requests by checking $\sigma_i \cdot P = \cdot H_1(PID_i)P_{pub} + PID_i^1 \cdot H_4(C_i' \parallel T_i)$ one by one (or using batch verification technology [39]),and aggregates the encrypted requests C_{sum} by computing Equation (2). Secondly, RSU calculates $\mu = (L(g^{\lambda}\ mod\ n^2))^{-1}\ mod\ n$, where function $L(x) = (x-1)/n$. Finally, RSU can get the frequency information $FI = \sum_{v=0}^{k} S_0(v) \parallel \sum_{v=0}^{k} S_1(v) \parallel ... \parallel \sum_{v=0}^{k} S_m(v) = L(C_{sum}^{\lambda}\ mod\ n^2) \cdot \mu\ mod\ n$.

It is worth noting that RSU can get the data downloading frequency information by FI. For example, assuming that CSP provides up to four kinds of services and RSU has received data downloading requests from eight vehicles that is as shown in Table III. After analyzing the encrypted requests from the eight vehicles, RSU can determine that the index of the most popular content are No.0 and No.1 content.

$$\begin{aligned} C_{sum} &= \prod_{i=0}^{k} C_i\ mod\ n^2 \\ &= \prod_{i=0}^{k} g^{(S_0(i) \| S_1(i) \| ... \| S_m(i))} x_i^n\ mod\ n^2 \\ &= g^{\sum_{i=0}^{k}(S_0(i) \| S_1(i) \| ... \| S_m(i))} \left(\prod_{i=0}^{k} x_i \right)^n\ mod\ n^2 \\ &= g^{(\sum_{i=0}^{k} S_0(i) \| \sum_{i=0}^{k} S_1(i) \| ... \| \sum_{i=0}^{k} S_m(i))} \left(\prod_{v=0}^{k} x_i \right)^n\ mod\, n^2 \end{aligned} \tag{2}$$

TABLE III
DOWNLOADING REQUEST SET OF THE EXAMPLE

Index i / Index v	0	1	2	3
0	000	001	000	000
1	001	001	000	000
2	001	001	000	000
3	001	000	000	000
4	000	000	000	000
5	001	001	000	000
6	000	001	000	000
7	001	001	001	000
$\sum_{i=0}^{7}(S_j(i))$	101	110	001	000

B. The Election Strategy of ECVs

After RSU knows which contents are the most popular, it can buffer these popular content to some vehicles to increase the success rate and efficiency of downloading high-frequency requested content. It is obvious that we can not cache these popular content to all nearby vehicles because of the limitation of wireless resource. So we need to choose some representative vehicles (ECVs) that have relatively better qualifications than ordinary vehicles. Since ECVs need to buffer some popular content, large storage space is the necessary requirement. Besides, vehicles that have lower speeds and are located in high-density area, bring a higher probability to transmit the data to nearby vehicles successfully. Traditional sorting methods are not suitable for ECVs election because sorting methods need to collect enough data which brings high latency. Luckily, fuzzy logic can solve the election problem timely and efficiently. To estimate the computation overhead brought by fuzzy logic control system, we have implemented the following fuzzification, fuzzy rules mapping, defuzzification processes using Scikit-fuzzy library [40], and the average computation overhead for one decision is about 1.5ms.

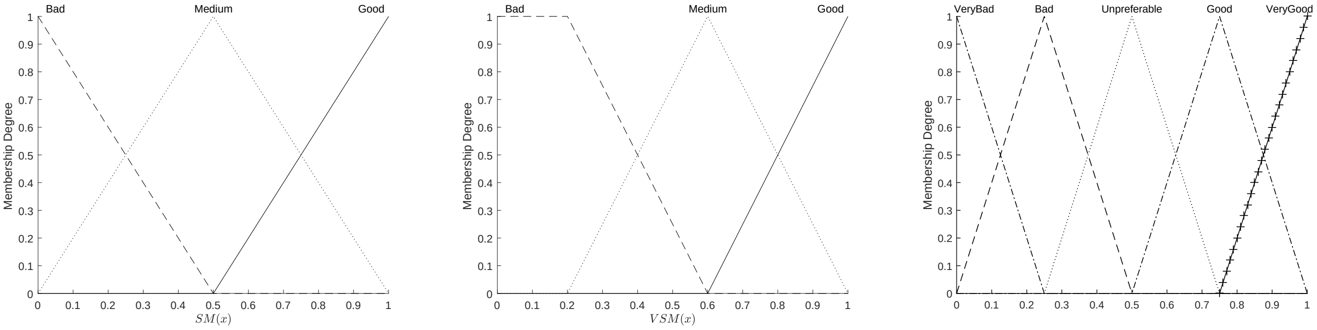
Fuzzification: We use the following membership function to convert the size of storage space and velocity to fuzzy values.

1) *Storage space factor*: ECVs need to cache some popular content which means that large storage space is the necessary requirement. We define storage space metric as Equation (3), where S denotes the predefined max storage space for caching popular content, $RSS(x)$ denotes the rest storage space size. The membership function of $RSS(x)$ is as shown in Fig. 4a.

$$SM(x) = \begin{cases} \frac{RSS(x)}{S}, & RSS(x) < S \\ 1, & RSS(x) \geq S \end{cases} \tag{3}$$

2) *Vehicle Service Stability*: ECVs, which are responsible for serving for nearby vehicles, should locate in the area where the communication quality with RSU is good and be slow because low speed provides better stability. We define vehicle service stability metric as Equation (4), where $V(x)$ denotes the velocity of the vehicle, and $l \in R$ means that the vehicle is in high-density area where the network channel contention is relatively frequent. The membership function of $RSS(x)$ is as shown in Fig. 4b.

$$VSM(x,l) = \begin{cases} \frac{1}{3} + \frac{2}{3} \times (1 - \frac{V(x)}{MAX_{y \in S} V(y)}), & l \in R \\ \frac{2}{3} \times (1 - \frac{V(x)}{MAX_{y \in S} V(y)}), & l \notin R \end{cases} \tag{4}$$



(a) SM membership function

(b) VSM membership function

(c) Output membership function

Fig. 4. SM, VSM, and output membership functions.

Fuzzy Rules: The fuzzy values calculated in *Fuzzification* step are mapped to linguistic values using IF-THEN rules and Min-Max method. And the fuzzy rules are as shown in Table IV.

TABLE IV
FUZZY RULES

Rule	Distance	Performance	Rank
Rule1	Short	Good	Very Good
Rule2	Short	Medium	Good
Rule3	Short	Bad	Unpreferable
Rule4	Medium	Good	Good
Rule5	Medium	medium	Unpreferable
Rule6	Medium	Bad	Bad
Rule7	Long	Good	Unpreferable
Rule8	Long	Medium	Bad
Rule9	Long	Bad	Very Bad

Defuzzification: In our scheme, the fuzzy result is defuzzified using Center of Gravity method which is calculated by $OV = \int \mu(x)x\,dx / \int \mu(x)\,dx$ and output membership function is shown in Fig. 4c. And the closer the output value OV is to 1, the more suitable the vehicle is to become ECV. We predefine a constant value k as a threshold. If $OV \geq k$, V_i will be elected as an ECV.

C. Downloading Process

Through the above two steps, some qualified vehicles become ECVs and the most popular data get determined. Hence RSU can send these popular data to nearby vehicles, and if any vehicle in the proximity of ECVs wants to download any one of these popular data, it can directly download the data from the nearby ECVs. It is worth noting that data provided by CSL are identified by indexes and these popular were transmitted to vehicles through RSU in non-accelerated stage, so RSU can directly find these popular data from its buffer memory or download the data again from CSP through high-bandwidth wired network.

VI. Security Proof and Analysis

A. Security Proof

Theorem 1. The ECC-based signature in our proposed scheme is secure against adaptive chosen message attacks.

Proof. Let $\mathcal{A}$ denote the adversary which is running in polynomial time against our proposed scheme in the random oracle model. The adversary $\mathcal{A}$ has a high authority to control the communications in VANETs. In particularly, $\mathcal{A}$ can monitor, modify and even fabricate messages. Let $\mathcal{C}$ denote the challenger who could solve the ECDLP problem with a non-negligible probability by running $\mathcal{A}$ as a subroutine. $\mathcal{C}$ simulates oracles which are queried by $\mathcal{A}$ as the following steps.

Setup: Firstly, the secure parameter n is chosen as the input of Setup algorithm. Secondly, $\mathcal{C}$ chooses the system private key s and calculates the corresponding system public key $P_{pub} = sP$. Finally, $\mathcal{C}$ sends public parameter set $\{P, q, G, E, P_{pub}\}$ to $\mathcal{A}$.

H_1 **hash query**: $\mathcal{C}$ produces a empty list L_{H_1}. After receiving the query with the message $\langle PID_i, T_i\rangle$ from $\mathcal{A}$, $\mathcal{C}$ checks whether the set $\langle PID_i, T_i, \tau_{H_1}\rangle$ is in the list L_{H_1} or not. If not, $\mathcal{C}$ chooses a number $\tau_{H_1} \in Z_q^*$ randomly, adds the set $\langle PID_i, T_i, \tau_{H_1}\rangle$ into the list L_{H_1} and sends $\tau_{H_1} = H_1(PID_i, T_i)$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ directly sends τ_{H_1} to $\mathcal{A}$.

H_2 **hash query**: $\mathcal{C}$ produces a empty list L_{H_2}. After receiving the query with the message M_i from $\mathcal{A}$, $\mathcal{C}$ checks whether the set $\langle Z, T_i, \tau_{H_2}\rangle$ is in the list L_{H_2} or not. If not, $\mathcal{C}$ chooses a number $\tau_{H_2} \in Z_q^*$ randomly, adds the set $\langle Z, T_i, \tau_{H_2}\rangle$ into the list L_{H_2} and sends τ_{H_2} to $\mathcal{A}$. Otherwise, $\mathcal{C}$ directly sends τ_{H_2} to $\mathcal{A}$.

Sign query: After receiving the query with message $M_i = \{PID_i, Z, T_i\}$ from $\mathcal{A}$, $\mathcal{C}$ first chooses four random numbers $\sigma, T_i, h_{i,1}, h_{i,2} \in Z_q^*$, a random point PID_i^2 in curve E, and calculates $PID_i^1 = (\sigma P - h_{i,2}P_{pub})/h_{i,1}$ to get $PID_i = \{PID_i^1, PID_i^2\}$. Next, $\mathcal{C}$ adds the set $\langle PID_i, T_i, h_{i,1}\rangle$ and $\langle Z_i, T_i, h_{i,2}\rangle$ into the list L_{H_1} and L_{H_2} respectively. And it is obviously that the above settings can ensure that the simulation scheme is indistinguishable from the real scheme.

After receives the responses from the oracles defined above, $\mathcal{A}$ outputs a message $\{PID_i, Z, T_i, \sigma_i\}$ and $\mathcal{C}$ checks the validity of the message by equation (1). According to Forking Lemma [18] [41] , $\mathcal{A}$ can output another valid message $\{PID_i, Z, T_i, \sigma_i\}$ if the above process is repeated with a different output of H_2 hash query, so that $\mathcal{C}$ can get two valid signatures $\sigma_i = sh_{i,1} + r_i h_{i,2}$ and $\sigma_i^{'} = sh_{i,1} + r_i h_{i,2}^{'}$ that can be used to calculate the value x by following equation:

$\frac{h_{i,2}^{'}\sigma - h_{i,2}\sigma^{'}}{h_{i,1}(h_{i,2}^{'} - h_{i,2})} \bmod q = s$

As a result, $\mathcal{C}$ can break the ECDLP within expected time less than $120686QT/\varepsilon$, where $\varepsilon \geq 10(R+1)(R+Q)/q$. However, it contradicts with the difficulty of solving the ECDLP. Therefore, the signature scheme in our scheme is secure against forgery under the alternatively chosen message attack in the random oracle model.

Theorem 2. If the MAC algorithm and ECDSA are both secure under adaptive chosen message attack, the CSL broadcast process of our scheme is secure.

Proof. There are three possible methods to attack the CSL broadcast process for malicious attackers. The first method is that the attacker may try to find a different message $m_i^{'}$ which has the same MAC with the original message, i.e., $MAC_{K_i}(m_i) = MAC_{K_i}(m_i^{'})$. However, it is in contradiction with the assumption that MAC algorithm is secure under adaptive chosen message attack. The second method is that the attacker may intend to get the undisclosed key in order to fabricate a message with a valid MAC. However, it is in contradiction with the one way property of hash function for which any adversary can not break the key in polynomial time. The third method is that the attacker may attempt to disguise as CSP to send a series of valid messages and MACs pair. However, CSP broadcast the packet which contains the initial key K_0 signed by its private key. As long as the private key of CSP is not disclosed, these messages and MACs pair will be discarded by vehicles because of the initial key check failure.

B. Security Analysis

In this section, we show that our proposed scheme satisfies several security requirements.

1) Message Authentication: In our proposed scheme, the CSL broadcasted by CSP is secured by TESLA algorithm. In Theorem 2, we have been shown that the authentication of CSL is guaranteed. And the authentication of messages sent from vehicles is guaranteed by ECC-based signing algorithm that is secure against adaptive chosen message attacks as shown in Theorem 1.

2) Request Privacy Preserving: After V_i determines its downloading request sequence RS_i, it sends different encrypted requests to CSP and RSU respectively. For the requests sent to CSP, it needs to be encrypted using AES encryption algorithm. As for the key for encryption, it is only known to V_i and CSP because of the computational difficulty of ECCDHP. For the encrypted requests sent to RSU, no one knows the original plaintext because the random number x_i is confidential. It is noteworthy that although RSU can not know the single original plaintext, it can still analyze the summary frequency information based on a heap of ciphertext due to the usage of additive homomorphic encryption.

3) Resistance to DDoS attack: One of the frequent communication scenario in our scheme is the CSL broadcast process which is carried out periodically. The attacker may attempt to produce many invalid message-and-MAC pairs to produce computation-based Denial of Service attack. However, the authentication of CSL is based on symmetric algorithm. The computation overhead of MAC and hash chain checking are very low. As for other frequent communications, the signature of the messages is ECC-based scheme which has very low computation overhead. Therefore, our scheme can resist DDoS attack to the relatively large extent.

4) Identity Conditional Privacy Preserving: In our proposed scheme, the identity contained in the message is vehicle's pseudo identity instead of real identity, because the real identity contained in multiple messages can be traced by malicious attackers which would cause the privacy leakage of the driver. The pseudo identity PID_i of vehicle contains two parts, i.e., PID_i^1 and PID_i^2, where $PID_i^1 = r_i \cdot P$ and $PID_i^2 = RID \oplus h\left(r_i \cdot P_{pub}\right)$. The attacker can not extract r_i in the case of known PID_i^1 and P because of the computational difficulty of discrete logarithm problem(DLP). Without knowing the secret key r_i, malicious attackers can not get the real identity RID from PID_i^1 and PID_i^2. However, TA should be able to trace the real identity of vehicle because some vehicles may send malicious messages to interfere with legal communication. In particular, TA can extract RID by PID_i^1 and PID_i^2 according to equation (5).

$$\begin{aligned} PID_i^2 \oplus h\left(s \cdot PID_i^1\right) &= RID \oplus h\left(r_i \cdot P_{pub}\right) \oplus h\left(s \cdot r_i \cdot P\right) \\ &= RID \oplus h\left(r_i \cdot P_{pub}\right) \oplus h\left(r_i \cdot P_{pub}\right) \\ &= RID \end{aligned} \tag{5}$$

5) Replay Attack Resistance: In order to resist replay attack, the message receiver needs to check the freshness of received message. Let T_R indicate the message receiving time, T_i indicate the timestamp contained in the message, Δt_1 indicate the clock difference, and Δt_2 indicate the network delay. The message will be received only if $|T_R - T_i| < \Delta t_1 + \Delta t_2$.

6) Resistance to Man-in-the-middle Attack: For the downloading request sent from vehicles to CSP, it is encrypted using AES encryption algorithm. Due to the security of AES (except for ECB mode), the attacker can't decrypt the encrypted request in polynomial time without knowing the session key. So the attacker may want to get the session key from the key agreement stage. However, the attacker can't get the session key because of the hardness of solving ECCDHP in polynomial time.

VII. Performance Analysis

A. Computation Overhead Analysis

In this subsection, we analyze the computation overhead of our proposed scheme brought by cryptographic operations. We measured the computation overhead of several core operation using MIRACL library [42] running on Intel i7-6700 processor and 8GB RAM. To measure the operation overhead, we construct an additive group generated by a point P on a non-singular elliptic curve $E : y^2 = x^3 + ax + b \bmod p$, and its order is q, where p,q are two 160 bit prime numbers, and $a, b \in Z_p^*$. Based on the settings, we have measured the average execution time of main cryptographic operations used in our scheme and found that the scale multiplication operation $x \cdot P$ of ECC (T_{sm-ecc}) takes about 0.442 ms, the scale exponential operation g^x of a 2048-bit finite field (T_{se}) takes about 2.126 ms, and the hash function (T_h) takes about 0.001 ms.

Using the measurements, the computation overhead of every part in our proposed scheme is as follows.

1) Vehicles: For our proposed scheme, the message receiver requires $n + 2$ scalar multiplication operations related to the ECC and n small scalar multiplication operations based on ECC to verify one message signature. Therefore, the computation overhead for signature verification is $(n + 2) \, T_{sm-ecc} + nT_{sm-ecc-s} = 0.4696n + 0.884$ ms. Next, we analyze the request encryption overhead. For the request sent to CSP, it needs to be encrypted based on AES-CBC encryption algorithm which costs less than 0.1ms, which is measured with Openssl library [43]. For the request sent to RSU, it needs to be encrypted using homomorphic encryption which requires one scalar multiplication, i.e., 0.442ms. Last, we analyze the CSL checking overhead. The main computation overhead of CSL checking process contains one ECDSA verification operation that costs 1.06 ms. It is noted that the computation overhead of hash chain process is very small, so we neglect it.

2) RSU: We assume that RSU has received n messages. For the message sent from vehicle, the single authentication process requires three scalar multiplication operations, which means that the total message authentication overhead is $1.326n$ ms. For the request aggregation process, n multiplication operations are required, which means that the request aggregation process overhead is $0.442n$ ms. For the aggregated requests decryption process, one exponential operation is required, which means that the aggregated requests decryption process overhead is 2.126 ms. Hence the total computation overhead of RSU is $1.326n + 0.442n + 2.126 = 1.768n + 2.126$ ms.

3) CSP: For the request message sent from the vehicle, the single authentication process requires three scalar multiplication operations which costs 1.326 ms. For the CSL generation process, one ECDSA signing operation is required which costs 2.623 ms. As for the secret key agreement phase, three scalar multiplication operations which cost 1.326 ms. Hence the total computation overhead of CSP is $1.326n + 1.4 + 1.326n = 2.652n + 1.4$ ms.

B. Communication Overhead Analysis

In this subsection, we analyze the communication overhead of our proposed scheme. We assume that the sizes of the output of the hash function and timestamp are 160 bits and 32 bits respectively. It is worth noting that the security strength of 160 bits ECC is considered equivalent to that of 1024 bits RSA cryptosystem. So an elliptic curve point in our scheme occupies 320 bits. The communication overhead of vehicle and CSP in our scheme are shown as follows respectively.

1) Vehicle: An ordinary message sent from the vehicle contains five parts: pseudo identity, encrypted request to RSU, encrypted request to CSP, timestamp and ECC signature. The pseudo identity PID_i of vehicle V_i contain two parts, i.e., PID_i^1 and PID_i^2 where $PID_i^1 \in G$, $PID_i^2 \in Z_q^*$. So the size of pseudo identity is $320 + 160 = 480$ bits. The size of the cipher related to homomorphic encryption in $Z_{n^2}^*$ equals to $2n$. Let the number of vehicle in the proximity of RSU be m and the amount of content contained in CSL be η. Hence the size of vehicle's request equals to $\lceil \eta log_2^m \rceil$ bit. If we choose the size of n to be 1024 bits, then the size of homomorphic encrypted request equals to 2048 bits. With PKCS7 [44], the size of request which is encrypted by AES algorithm equals to $128\lceil(\lceil \eta log_2 m \rceil)/128 + 1\rceil$ bits. The sizes of timestamp and ECC signature equal to 32 bits and 320 bits respectively.

2) CSP: The broadcast content of CSP mainly contains initial key broadcast and CSL broadcast. The message for initial key broadcast is $\{T_i, K_0, ECDSA_{sign}(K_0)\}$ that occupies $32 + 160 + 512 = 704$ bits. The security part of message for CSL broadcast is $\{T_i, MAC_{K_i}(CSL \parallel T_i), K_{i-1}\}$ that occupies $32 + 160 + 160 = 362$ bits.

C. Experiment Analysis

In order to evaluate the network performance of our proposed scheme, we use three tools: OMNeT++ [45], Sumo [46] and Veins 4.6 [47]. OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators, which supports the simulation for wired network and wireless ad-hoc network [48]. Sumo is an open-source, highly portable, microscopic and continuous road traffic simulation package designed to handle large road networks. Veins acts as a middleware between Omnet++ and Sumo. The network parameters are as shown in Table V. To assess the network performance of our scheme, we consider two metrics: packet loss ratio and average message transmission delay.

TABLE V
SIMULATION PARAMETER SETTINGS FOR OUR SCHEME

Parameters	Values
Simulation Area	$2500 \times 2500(m^2)$
Path Loss Model	Two-Ray Interference Model
Obstacle Shadowing Model	Simple Obstacle Shadowing Model
Max Interference Distance	2600m
Transmission Power	20mW
Data Transmission Rate	6Mbps
Sensitivity	-89dBm
Thermal Noise	-110dBm
Beacon Interval	1s
Downloading Request Interval	5s
Used Channel	CCH
Max speed	40m/s
Max Number of Vehicle	50
Acceleration	$10m/s^2$
Simulation Time	200s

1) Packet loss ratio: The packet loss ratio is defined as the ratio between the number of messages dropped and the number of messages received by vehicle, which can be expressed as Equation (6) where $AVG(\cdot)$ denotes averaging function, n denotes the number of vehicles, $N^i_{received}$ denotes the number of messages received in MAC layer by vehicle V_i and N^i_{lost} denotes the number of lost messages in MAC layer by vehicle V_i.

$$PLR = AVG(\sum_{i=1}^{n} \frac{N^i_{lost}}{N^i_{received} + N^i_{lost}}) \tag{6}$$

Firstly, we analyze the relationship between the packet loss ratio and the size of content provided by CSP. As we know, the core task of a vehicle is to broadcast beacons or BSMs (Basic Safety Messages) to nearby vehicles to improve road safety in VANETs. However, data downloading service may disturb regular beacon transmission, because large data downloading may occupy wireless channel in a long time and then cause a high packet loss ratio. We compare the packet loss ratio with Hao et al.'s scheme [11] and Lai et al.'s scheme [12], which are two representative schemes in secure data downloading/sharing research area, as shown in Fig. 5a. From Fig. 5a, we can get a conclusion that our proposed has a lower packet loss ratio. The reason why our scheme has a lower packet loss ratio is that vehicles retrieve content from nearby ECVs directly in our scheme. But in Lai et al.'s scheme [11], the vehicle which has data downloading request needs to download the data from several proxy vehicles and relay vehicles in a chain which could cause frequent channel contention.

Secondly, we analyze the relationship between packet loss ratio and average speed of vehicle and the experimental results are as shown in Fig. 5b. From Fig. 5b, we can observe two facts: 1) The three schemes have similar trends and appropriate speed can bring relatively low packet loss ratio; 2) More vehicles involvement which aims to increase the downloading success rate brings higher packet loss ratio for vehicles.

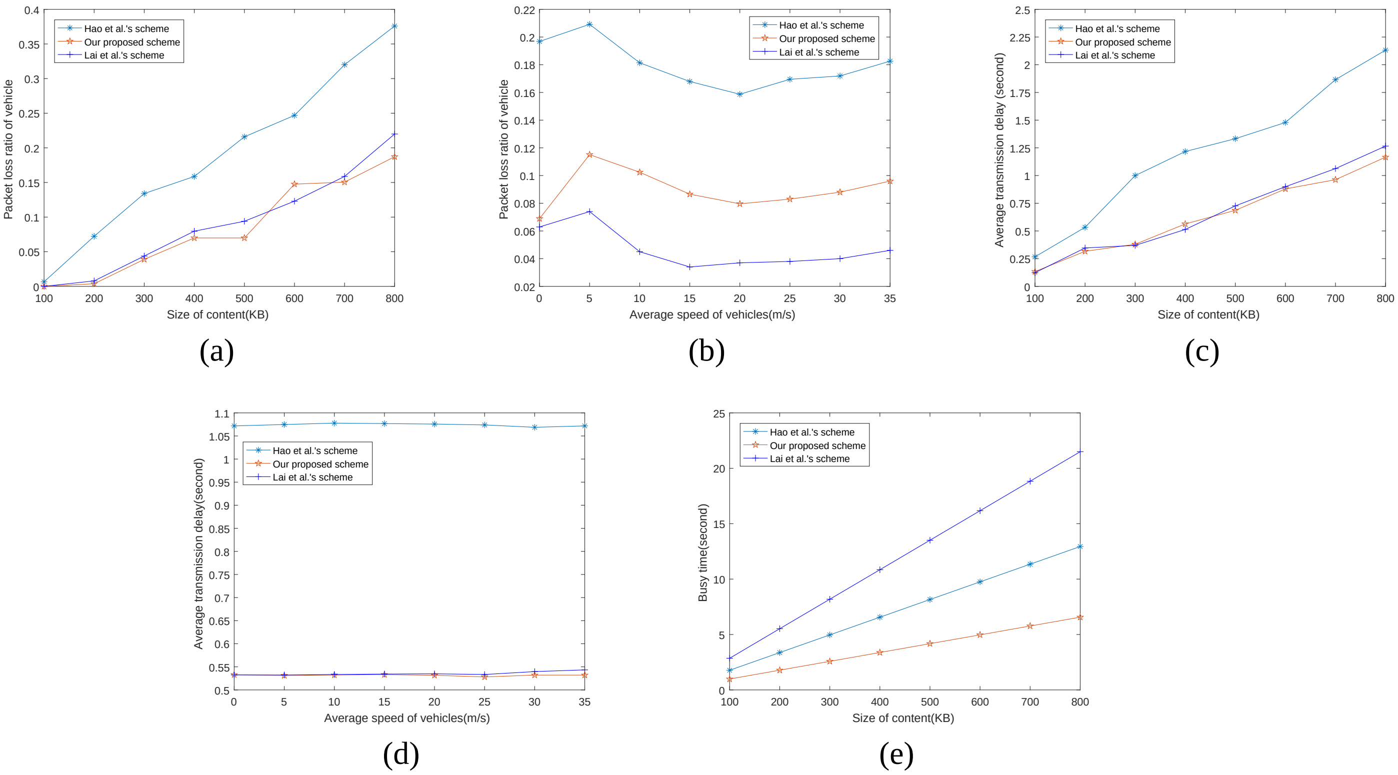
2) Transmission delay: The average transmission delay is defined as the average time cost to transmit the message from the sender to the receiver, which can be expressed as Equation (7) where n denotes the number of vehicles, N_i denotes the number of messages received by vehicle V_i, $T^j_{received}$ denotes the receiving time of message m_j by receiver and T^j_{send} denotes the sending time of message m_j by sender. Fig. 5c shows the relationship between average transmission delay and the size of the content provided by CSP. We can find that the transmission delay in our scheme is lower than that of the other two schemes. Fig. 5d shows the relationship between the average transmission delay and average speed of vehicles. We can observe that the degree that the speed of vehicle affects transmission delay is small.

$$ATD = \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{N_i}\sum_{j=1}^{N_i}(T^j_{received} - T^j_{send})) \tag{7}$$

3) MAC layer busy time: As we know, the number of RSUs is always limited which means that the burden on RSU is high. The MAC layer's busy time reflects the degree of busyness. Fig. 5e shows the relationship between the busy time of MAC layer of RSU and the size of content provided by CSP. From Fig. 5e, we can observe that the MAC layer busy time and the size of content provided by CSP approach a linear relationship approximately, and our scheme brings the least burden on RSU compared with Hao et al.'s and Lai et al.'s schemes.

VIII. Conclusion

Social media applications and real traffic information produce large amounts of data, which introduces the data downloading scenario in VANETs. However, simply downloading from the infrastructure is not quite efficient. In this paper, we have proposed a secure and efficient data downloading scheme based



(a) (b) (c) (d) (e)

Fig. 5. (a) The relationship between packet loss ratio and the size of content provided by CSP, where the max speed of vehicle is 20m/s. (b) The relationship between packet loss ratio and average speed of vehicle, where the size of requested data is 400KB. (c) The relationship between average transmission delay and the size of content provided by CSP, where the max speed of vehicle is 20m/s. (d) The relationship between average transmission delay and average speed of vehicle, where the size of requested data is 400KB. (e) The relationship between MAC layer busy time of RSU and the size of content provided by CSP.

on edge computing. The RSU analyzes the data request frequency and determines the most popular data. Further, the RSU caches these data in nearby ECVs, which are selected using a fuzzy-logic-based strategy. If any vehicle wishes to download these popular data, it can be directly done from the nearby ECVs. This method can increase the resource downloading efficiency without sacrificing the privacy of a vehicles download request. Security analysis and simulation results demonstrated that the proposed scheme could resist malicious attacks along with a good network performance.

In the future, we would work toward solving various issues in secure cooperative downloading research areas. Firstly, we will design a lightweight and cryptography-based incentive mechanism to stimulate cooperative downloading. Secondly, we will use dynamic allocation technology to solve the data allocation issue of the cooperative downloading process. Thirdly, we will analyze the secure cooperative downloading issue in 5G-based VANETs.

Acknowledgment

The work was supported by the National Natural Science Foundation of China (No. 61872001, No. 61702005), the Key Program of National Natural Science Foundation of China (No. U1936220), the Open Fund of Key the Key Program of National Natural Science Foundation of China (No. U1936220), the Open Fund of Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education (No. ESSCKF2018-03), the Open Fund for Discipline Construction, Institute of Physical Science and Information Technology, Anhui University and the Excellent Talent Project of Anhui University. The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this paper.

References

[1] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong, "An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1621–1632, 2018.

[2] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "Pa-crt: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[3] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[4] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2722–2730.

[5] J. Ott and D. Kutscher, "Drive-thru internet: Ieee 802.11b for "automobile" users," in *Joint Conference of the IEEE Computer and Communications Societies*, 2004.

[6] W. Saad, H. Zhu, A. Hjorungnes, D. Niyato, and E. Hossain, "Coalition formation games for distributed cooperation among roadside units in vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 48–60, 2011.

[7] H. Liang and W. Zhuang, "Cooperative data dissemination via roadside wlans," *Communications Magazine IEEE*, vol. 50, no. 4, pp. 68–74, 2012.

[8] H. Zhou, B. Liu, T. H. Luan, F. Hou, L. Gui, Y. Li, Q. Yu, and X. Shen, "Chaincluster: Engineering a cooperative content distribution framework for highway vehicular communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2644–2657, 2014.

[9] Y. Hao, J. Tang, Y. Cheng, and C. Zhou, "Secure data downloading with privacy preservation in vehicular ad hoc networks," in *2010 IEEE International Conference on Communications*. IEEE, 2010, pp. 1–5.

[10] R. Mishra, A. Singh, and R. Kumar, "Vanet security: Issues, challenges and solutions," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. IEEE, 2016, pp. 1050–1055.

[11] Y. Hao, J. Tang, and Y. Cheng, "Secure cooperative data downloading in vehicular ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 523–537, 2013.

[12] C. Lai, K. Zhang, N. Cheng, H. Li, and X. Shen, "Sirc: A secure incentive scheme for reliable cooperative downloading in highway vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–16, 2017.

[13] N. Magaia, Z. Sheng, P. R. Pereira, and M. Correia, “Repsys: A robust and distributed incentive scheme for in-network caching and dissemination in vehicular delay-tolerant networks,” *IEEE Wireless Communications*, vol. 25, no. 3, pp. 65–71, 2018.

[14] J. Zhang, Q. Zhang, and W. Jia, “Vc-mac: A cooperative mac protocol in vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1561–1571, 2009.

[15] M. Raya and J.-P. Hubaux, “Securing vehicular ad hoc networks,” *Journal of computer security*, vol. 15, no. 1, pp. 39–68, 2007.

[16] C. Zhang, X. Lin, R. Lu, P.-H. Ho, and X. Shen, “An efficient message authentication scheme for vehicular communications,” *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3357–3368, 2008.

[17] T. W. Chim, S.-M. Yiu, L. C. Hui, and V. O. Li, “Specs: Secure and privacy enhancing communications schemes for vanets,” *Ad Hoc Networks*, vol. 9, no. 2, pp. 189–203, 2011.

[18] D. He, S. Zeadally, B. Xu, and X. Huang, “An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[19] J. Cui, J. Zhang, H. Zhong, and Y. Xu, “Spacf: A secure privacy-preserving authentication scheme for vanet with cuckoo filter,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 283–10 295, 2017.

[20] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher, “Cuckoo filter: Practically better than bloom,” in *ACM International on Conference on Emerging NETWORKING Experiments and Technologies*, 2014, pp. 75–88.

[21] H. Zhong, B. Huang, J. Cui, Y. Xu, and L. Liu, “Conditional privacy-preserving authentication using registration list in vehicular ad hoc networks,” *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2017.

[22] A. Dua, N. Kumar, A. K. Das, and W. Susilo, “Secure message communication protocol among vehicles in smart city,” *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2018.

[23] A. Perrig, “The tesla broadcast authentication protocol,” *Rsa Cryptobytes*, vol. 20, no. 2, p. 2002, 2002.

[24] A. Studer, F. Bai, B. Bellur, and A. Perrig, “Flexible, extensible, and efficient vanet authentication,” *Journal of Communications and Networks*, vol. 11, no. 6, pp. 574–588, 2009.

[25] 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.

[26] L. Chen, D. Gu, Y. Zeng, and P. Mohapatra, “Pba: Prediction-based authentication for vehicle-to-vehicle communications,” *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 1, pp. 71–83, 2016.

[27] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[28] D. Miao, L. Liu, R. Xu, J. Panneerselvam, Y. Wu, and W. Xu, “An efficient indexing model for the fog layer of industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, pp. 4487–4496, January 2018.

[29] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, “Optimal delay constrained offloading for vehicular edge computing networks,” in *IEEE International Conference on Communications*, 2017, pp. 1–6.

[30] C. M. Huang, M. S. Chiang, D. T. Dao, W. L. Su, S. Xu, and H. Zhou, “V2v data offloading for cellular network based on the software defined network (sdn) inside mobile edge computing (mec) architecture,” *IEEE Access*, vol. 6, no. 99, pp. 17 741–17 755, 2018.

[31] J. Feng, Z. Liu, C. Wu, and Y. Ji, “Ave: Autonomous vehicular edge computing framework with aco-based scheduling,” *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[32] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, “Raise: An efficient rsu-aided message authentication scheme in vehicular communication networks,” in *Communications, 2008. ICC’08. IEEE International Conference on*. IEEE, 2008, pp. 1451–1457.

[33] V. Miller, “Use of elliptic curves in cryptography,” in *Advances in CryptologyłCRYPTO85 Proceedings*. Springer, 1986, pp. 417–426.

[34] N. Koblitz, “Elliptic curve cryptosystems,” *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[35] P. Paillier, “Public-key cryptosystems based on composite,” *Eurocrypt*, vol. 547, no. 1, pp. 223–238, 1999.

[36] P. Hjek, “Metamathematics of fuzzy logic,” *Trends in Logic*, vol. 4, pp. 155–174, 1998.

[37] W. Pedrycz, *Fuzzy control and fuzzy systems (2nd, extended ed.)*. Research Studies Press Ltd., 1993.

[38] X. Bai, L. Liu, M. Cao, J. Panneerselvam, Q. Sun, and H. Wang, “Collaborative actuation of wireless sensor and actuator networks for the agriculture industry,” *IEEE Access*, vol. 5, pp. 13 286–13 296, 2017.

[39] J. Camenisch, S. Hohenberger, and M. Ø. Pedersen, “Batch verification of short signatures,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2007, pp. 246–263.

[40] “Scikit-fuzzy: A fuzzy logic toolbox for scipy,” https://github.com/scikit-fuzzy/scikit-fuzzy, accessed 1 Feb, 2019.

[41] D. Pointcheval and J. Stern, “Security arguments for digital signatures and blind signatures,” *Journal of cryptology*, vol. 13, no. 3, pp. 361–396, 2000.

[42] M. Scott, “Miracl–multiprecision integer and rational arithmetic c/c++ library,” *Shamus Software Ltd, Dublin, Ireland*, 2003.

[43] “Openssl: Tls/ssl and crypto library,” https://github.com/openssl/openssl, accessed 1 Feb, 2019.

[44] B. Kaliski, R. Laboratories, B. S. Kaliski, R. Laboratoties, and K. W. Kingdon, “Extensions and revisions to pkcs 7,” 1997.

[45] “Omnet++ decrete event simulator,” Available from:https://www.omnetpp.org, accessed July 1, 2018.

[46] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “Sumo - simulation of urban mobility: an overview, simul 2011,” in *Simul*, 2011, pp. 63–68.

[47] C. Sommer, R. German, and F. Dressler, “Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, January 2011.

[48] G. Min, Y. Wu, and A. Y. Al-Dubai, “Performance modelling and analysis of cognitive mesh networks,” *IEEE Transactions on Communications*, vol. 60, no. 6, pp. 1474–1478, 2012.