Hindawi
Security and Communication Networks
Volume 2022, Article ID 9779670, 11 pages
https://doi.org/10.1155/2022/9779670

*Research Article*

# A Lightweight Authentication with Dynamic Batch-Based Group Key Management Using LSTM in VANET

**Xieyang Shen**,[1,2] **Chuanhe Huang**,[1,2] **Wenxin Pu**,[1,2] **and Danxin Wang**[1,2]

[1]*School of Computer Science, Wuhan University, Wuhan, China*
[2]*Collaborative Innovation Center of Geospatial Technology, Wuhan, China*

Correspondence should be addressed to Chuanhe Huang; huangch@whu.edu.cn

Received 22 August 2021; Accepted 4 January 2022; Published 3 March 2022

Academic Editor: Gu Zhaoquan



Due to its complexity and mobility, VANET (vehicle ad hoc network) security has long plagued the development of the IoT industry. It is still a big challenge for users to decide the trustworthiness of an anonymous message or the preservation of personal information. Group signature is widely used in VANET anonymous authentication, but the existing solutions suffer from high computation costs in certificate revocation list (CRL) checking and signature verification process. In our scheme, we develop a lightweight protocol based on hashing functions and group keys, which escapes from the heavy computation cost. Then, we propose a dynamic batch-based group key distribution process, which is based on long short-term memory (LSTM) neural network to predict traffic flow and calculate the weight to determine the right time for key update. In this way, our method will significantly reduce computation delay and communication overhead. The security and performance analyses show that our scheme is more efficient in terms of authentication speed while keeping conditional privacy in VANET.

## 1. Introduction

As a critical component of the intelligent transportation system [1], VANETs' main goal is providing safety assurance and comfort service for passengers [2, 3]. Vehicles form a particular type of network in which they have no fixed position. Besides, they can authenticate other vehicles around them to form a network and do some necessary communications. High mobility of nodes is the main feature of such networks enabling nodes to frequently change their pattern. These kinds of rapid changes bring many network security issues. Considering the insecurity in VANET, designing a secure communication solution is the most urgent challenge in this field [4].

A VANET security model mainly consists of three components, TA (trusted authority), RSU (roadside unit), and OBU (onboard unit). The TA provides internet connectivity and stores important information such as the real identity of all the vehicles and RSUs. TA is also responsible for generating public parameters. RSUs are distributed along the road and used to manage OBUs within their communication range. OBU is a temper-proof device, which is attached to a vehicle to help communicate with other infrastructures through wireless communication. Figure 1 shows the VANET communication pattern.

Information exchange in VANET is ongoing all the time between vehicles and infrastructure such as alarm signals, traffic information, weather conditions, multimedia material, or any other kind of data. Besides, applications on VANET can provide passengers convenience, safety information, and other attractive features. However, security issues bring lots of concerns as it is hard to balance the demand of low latency and high security. The insecurity caused by serious consequences of cyberattacks will incur countless threats, and vulnerabilities due to high mobility network feature rapidly changing network topology and an open environment. As such, a secure scheme must be proposed with robust and reliable security measurements to prevent malicious activities and preserve users' privacy [5].

*1.1. Security Requirement.* There are several attributes in VANET security [6] including but not limited to the following:

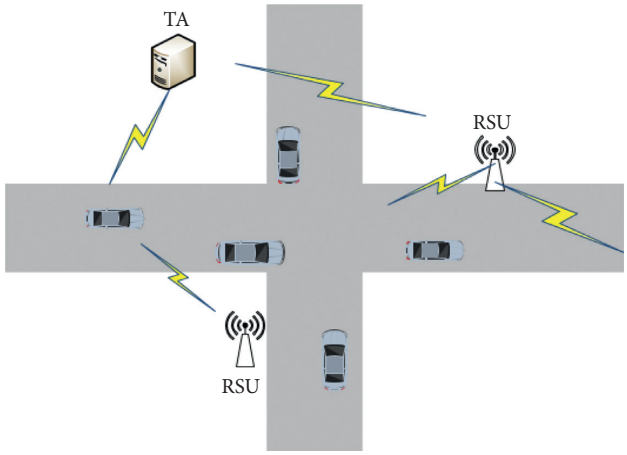


Figure 1: VANET communication.

(i) Message authentication: Authentication ensures that a message is trustable. If the message spread in VANET has not been authenticated, the users cannot judge the traffic situation or make wrong decisions.

(ii) Privacy preserving: It is, perhaps, the most critical security thing to protect personal security. The leakage of personal information may result in malicious crimes. In VANET, nodes are communicating in a public channel. Therefore, anonymity must be guaranteed.

(iii) Traceability: While hiding the real identity of the user, the TA or some other trusted authority should have the ability to reveal the real identity of the nodes. If a malicious node sends a fake message and leads to accidents, this malicious node should be identified by a trusted authority. That is also an essential issue in group signature.

In VANET, we should have the conditional privacy of the nodes, that is, the combination of the vehicle's privacy preservation and its traceability. When a vehicle enters an area, if it has a desire to send some message or request some service, it should keep anonymity at first. There are many schemes to guarantee anonymity, for example, schemes based on a large number of anonymous keys or based on a unique identity.

The group signature [7] is an anonymous authentication scheme that forms a group with a set of users, while the users remain anonymous to each other. The group incorporates multiple group private keys with one group public key, called "group key." Each group member can anonymously sign messages on behalf of the whole group, and the real identity can be revealed by the group manager. Therefore, the group signature can effectively achieve conditional privacy preservation for VANET communication [8].

Although group signature is widely used in VANET to realize anonymous authentication, the existing solutions suffer from long computation delays in the signature verification process and CRL checking. This problem is severe when the CRL becomes very large. [9]. The CRL holds all the revoked anonymous keys. The increase in anonymous keys makes the CRL volume becomes large, which significantly increases the time of authentication. Because that before verifying the signature, vehicles should verify a large CRL to check whether the signer is revoked or not. As a result, these schemes cannot meet the requirement of verifying a large number of messages in VANET. In the literature, many revoking methods have been proposed, perhaps we can use an ID-based method to avoid the revoking process. In addition, many authentication schemes use bilinear-pairing or elliptic curves for implementation. The computation pressure sometimes can be very high. We should have a robust and lightweight protocol to fit the large-scale network environment.

In urban areas, if a vehicle wants to request some service like weather conditions, traffic situations, or deliver some information with surrounding nodes, it can form a group with other vehicles. They share the same group key at a specific time. Only the legal members can preserve the group key, and in this way, they can have a trusted relationship. In urban areas with a large density of vehicles, vehicles frequently enter and exit the area. Many protocols update the group key whenever a vehicle comes in and out of the area. If the calculation of the updating phase is not that novel and efficient, the calculation stress and the frequent communication overhead will bring much pressure. So we need a novel solution that handles the overhead and efficiency at the same time. Furthermore, we do not consider the periodical beacon message, and group signatures generally do not apply to this type of message but are suitable for slightly longer communication needs. These communications are ideal for requesting services and passing personalized information.

As stated above, the RSU registers with the TA in its range at first, and a RSU usually belongs to only one TA's range. Vehicles also register to TA. Next, if the vehicle does not have urgent communication desire, it will wait to get a new group key in a time slot. If it has a strong desire to send messages, it sends a request to RSU. Then, RSU will judge the urgency of messages and decide whether to trigger a new group key update at once. So we have a dynamic adjustment strategy.

*1.2. Contributions.* ·We propose a lightweight authentication protocol without any complex computation and also a simple group key generation and verification process. In our scheme, RSU plays the role of generating group keys. We can achieve both anonymity and traceability.

·We design a dynamic batch-based group key updating method to reduce the communication cost and the frequency of the group key updating phase. Our method is based on priority, which can fit the need of urgent communication and reduce communication overhead.

*1.3. Structure.* Section 2 describes the related works, and section 3 provides a detailed explanation of the lightweight authentication and group key generation. Section 4 explains the batch-based group key update phase. Section 5 is the

security analysis and performance comparison. Section 6 concludes the study.

## 2. Related Work

Vehicle authentication schemes can be generally divided into two categories, and there are certificate-based protocols that use public and private key pairs, and ID-based protocols, where the user's public key can be computed from the user's identity (i.e., e-mail address and IP address). However, TA and RSUs must keep all the public and private key pairs of vehicles. Each vehicle also needs a large storage space to store the public and private key pairs [10]. Zhang et al. [11, 12] took advantage of the ID-based protocol and proposed two ID-based privacy-preserving authentication protocols. The ID-based authentication protocols found in [13] deal with the need for complex certificate management. However, these protocols still have high computation costs.

In the literature, we have known that the earlier certificate-based protocols store the public and private key pairs of the vehicles. They also need an efficient public and private key management solution to manage, distribute, and revoke all public key certificates. The ID-based protocols are considered to overcome the problems of certificate-based protocols. However, these protocols are usually computationally complex and time-consuming, because these protocols are implemented either using the elliptic curve [10] or the bilinear pairing [13]. The bilinear pairing is known as a time-consuming operation when compared to other cryptographic functions like hashing and elliptic curve operations that are costly too.

Chaum et al. and van Heyst [7] introduced a group signature for anonymous authentication, which uses several group private keys attached to one group public key. Under this scheme, the users can verify the validity of the group signature without knowing the real identity of other group members, and the real identity can be revealed when needed. Any pair of signatures created by the same group user cannot be linked by any third party except the legal group manager, and group members can verify any signed message using the group public key.

Although group signature is an excellent technique for VANET's privacy preservation, it still suffers from large computation overhead in the signature verification process. The short group signature (SGS) scheme was introduced by Boneh et al. [14], which is one of the most important GS schemes in the literature. Lin et al. [15] implement SGS and identity-based signatures for securing VANETs, in which all the OBUs form a group and have a unique private key to sign the messages. Lu et al. [16] used group signatures with the RSU, where each RSU uses its group private key to update the short lifetime certificate of the OBUs. Calandriello et al. [17] use group signatures at the OBU level, where OBUs use their group private keys for signing short lifetime certificates for themselves. The public keys in the generated short lifetime certificates are used to sign the outgoing messages. In [18], the trusted authority (TA) generates and manages the group keys. In [19], the TA is designed to classify the users and distribute the group keys to the group of users, and keys will be updated during the revocation process. These schemes will lead to a large load on the trusted authority. The above group signature schemes neglect the significant computation overhead embedded in them.

In group communications, a novel protocol is required to generate and manage a group key that can be used to secure data sent from group members to all users that are members of the same group. Multicast groups are very dynamic, because of the joining of new members and the leaving of old members, and the group key manager has to handle group membership changes by regenerating and redistributing new group keys. Generally, after each membership change, the group key should be updated through an appropriate rekeying phase, such that a new vehicle cannot compute the old group key (backward secrecy [19]), and a leaving vehicle cannot compute the new group key (forward secrecy [19]). In many schemes, the group key is refreshed immediately after any join or leave events, and such events' number is often proportional to the number of changes in membership events. If the updating phase frequently happens in a congested urban area, the group key may have expired when it has not been used yet. These operations may cause many communication costs in vain. Thus, a batch process algorithm must be proposed according to the traffic flow.

Artificial intelligence has played an important role in many scenarios such as weather reporting [20], game strategy [21], and cognitive radionetworks [22]. Long short-term memory (LSTM) is introduced as the nonlinear dynamic soft-sensing method for predicting traffic flow [23]. First, multiple consecutive real-time samples of a batch process are used as the query samples. During the similarity calculation stage, three similarity measurements are adopted including the information of the distance, angle, and trend to take the traffic rendezvous into consideration [24]. For each individual measurement, historical trajectories with larger similarity measurements are collected as the online modeling samples. Hence, several LSTM soft sensor models can be constructed with the extracted batch trajectories and used for the quality prediction of online query samples. To integrate the quality prediction results of different submodels, weighting parameters of different similarity measurements are defined and calculated based on the cross-validation strategy. Finally, the weighted sum of each prediction result is judged as the ensemble result of the real-time batch trajectory.

Another important security thing is that if there is a malicious node that does not send the timely message to key generator, it can secretly keep this key for a long time. In fact, it has already left this region or delivered this key. These kinds of problems can also be found in data access control schemes [25]. Many protocols have not solved this problem, and their leaving updating phase cannot actually judge the vehicle leaving time. They usually depend on honest vehicles to send leaving messages to key generator, which cannot prevent malicious nodes.

From the above state-of-the-art solutions, we could see that verification in VANET still faces the problem of efficiency. Besides, due to the uncertainty of the vehicles, it is

also hard to use batch authentication to reduce the time cost. Another drawback is the bottleneck on TA for most schemes that need TA to undertake most of the computation works to generate keys.

We propose a scheme tailored for very dynamic ad hoc networks. Time is dynamically partitioned in any length slot. Each slot has a unique group key, and although users asynchronously join, the group manager will decide the beginning of the time slot according to users' message priority. Although this kind of method introduces a delay, it also allows to reduce the number of rekeying acts. In our scheme, RSUs play the role of the group manager, which can release the burden of TA and reduce communication costs between TA and RSUs. We assume that RSU is equipped with the highly trusted platform modules since it is a key component in the key generation and management processes.

## 3. Lightweight Protocol

It is known that certificate-based protocol requires lots of storage at TA and vehicles, and a complex certificate and key management process. ID-based protocols do not need to do this, but they are often time-consuming. Since they either use bilinear pairing or the elliptic curve, which is known as time-consuming, in our protocol, we use the lightweight hash function to overcome these problems, which can also provide the same security.

The general cryptographic hash function is lightweight and is often considered impracticable to reverse. If we only know the output of a hash function, it is infeasible to compute the input value of that hash function. Even if a slight change in input has occurred, the output value will have a big difference. So our protocol has little computation cost, since operations like XORing and hashing are very lightweight.

Our protocol is divided into seven phases: 1, System initialization; 2, RSU registration phase; 3, Vehicle registration phase; 4, Vehicle authentication phase; 5, Group key generation phase; 6, Vehicle joining phase; 7, Vehicle leaving phase.

Figure 2 provides an overview of the authentication process.

The notations in Table 1 are used in our protocol.

*3.1. System Initialization.* In our protocol, TA is considered fully trusted and initializes all the system's parameters. During the registration phase, TA delivers these parameters to RSUs and vehicles. The initialization phase is as follows.

(1) TA selects a large prime number $q$ and a finite field $Z_q^*$.

(2) TA selects a secure hash function H, where H: $\{0,1\}^* \longrightarrow \{0,1\}^l$.

*3.2. RSU Registration.* The $RSU_i, i \in \{1,2,3,\ldots,n\}$, sends the information about itself to which it is securely belonged to the TA. TA then gives a unique identity $ID_{r_i}$ and two secret keys $SK_{r_i}$ and $Sv_{r_i}$ to the $RSU_i$ through a secure channel, which is usually wired. The $SK_{r_i}$ is a shared key between the TA and the $RSU_i$, and the $Sv_{r_i}$ is used for the $RSU_i$ and vehicle's communication. So RSU gets $\{Z_q^*, q, H, SK_{r_i}, Sv_{r_i}, ID_{r_i}\}$ from TA.

*3.3. Vehicle Registration*

(1) The $V_j$, $j \in \{1,2,3,\ldots,m\}$ registers to the TA.

(2) The $V_j$ selects a unique identity $ID_{v_j}$, which is associated with its real identity, such as license plate, and then securely sends it to TA.

(3) TA then sends $Z_q^*, q, H, Q = H(Sv_{r_i})$ to $V_j$ through a secure channel, and $Q$ is a list, which is used for vehicle to check the RSUs' trustworthiness. These RSUs are within this TA's region.

*3.4. Vehicle Authentication Phase.* When a vehicle enters an RSU's domain, it should send an authentication message to RSU to prove that he is legal and get some regional-related message. Only the vehicle has passed the authentication, and it can get the group key.

The vehicle's authentication with RSU has the following steps:

(1) The vehicle $V_j$ chooses a random number $s_{v_j}$ as his secret, computes $A_{v_j} = H(ID_{v_j} \| s_{v_j})$, and then sends $A_{v_j}$ to RSU in a public channel.

(2) The RSU chooses a random number $c_{r_i}$, then computes $R_{v_j} = H(Sv_{r_i} \| c_{v_j})$, $M_{v_j} = A_{v_j} \oplus R_{v_j}$, $N_{v_j} = H(R_{v_j} \| A_{v_j} \| c_{v_j})$, and sends $M_{v_j}, N_{v_j}, c_{v_j}$ to $V_j$.

(3) Once $V_j$ receives the message, and it calculates $R_{v_j}^* = M_{v_j} \oplus A_{v_j}$, $N_{v_j}^* = H(R_{v_j}^* \| A_{v_j} \| c_{v_j})$, to verify the RSU's message. If the $N_{v_j}^*$ does not match the $N_{v_j}$, the message might be modified, and the vehicle refuses the message.

(4) Then, $V_j$ computes $VID_{v_j} = ID_{v_j} \oplus H(R_{v_j}^* \| Tv_j)$, in which $Tv_j$ is the current timestamp. The $V_j$ chooses a random number $a_{v_j}$ and computes $b_{v_j} = a_{v_j} \oplus H(R_{v_j}^* \| VID_{v_j})$, $Dv_j = H(R_{v_j}^* \| ID_{v_j} \| Tv_j)$.

(5) $V_j$ sends $O_{v_j} = \{VID_{v_j}, c_{v_j}, b_{v_j}, Dv_j, Tv_j\}$ to $RSU_i$ through a public channel.

(6) When $RSU_i$ received the message, it can compute $R_{v_j} = H(Sv_{r_i} \| c_{v_j})$ and $ID_{v_j}^* = VID_{v_j} \oplus H(R_{v_j} \| Tv_j)$ and also compute $Dv_j^* = H(R_{v_j} \| ID_{v_j} \| Tv_j)$, if $Dv_j^* = Dv_j$, and the massage has not been modified. Then, the $RSU_j$ calculates $S_{r_i} = En_{SK_{ri}}(ID_{v_j}^*)$, which means $ID_{v_j}^*$ is encrypted with $SK_{r_i}$. $SK_{r_i}$ is a shared key between $RSU_i$ and TA. They communicate with each other in a secure channel.

(7) The TA checks the $ID_{v_j}$ registered list, if it finds $ID_{v_j} = ID_{v_j}^*$, and then returns true to RSU; otherwise, it

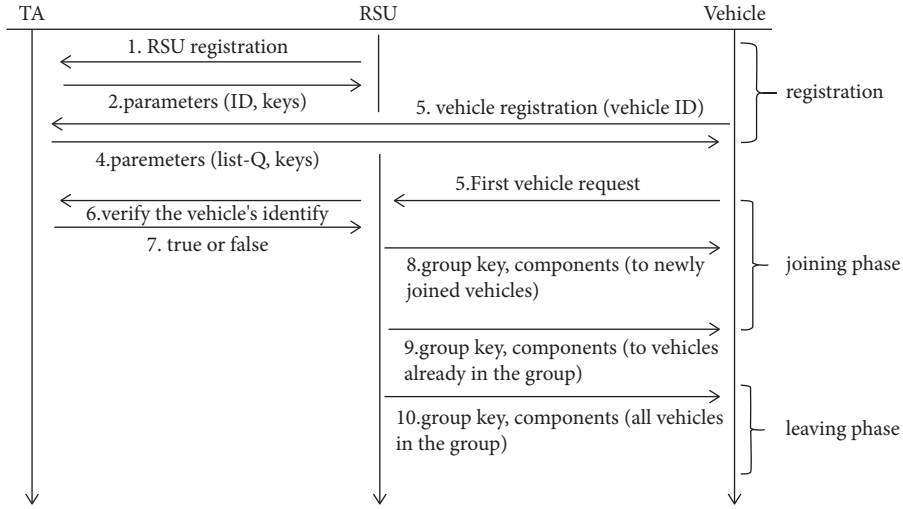


Figure 2: Authentication phase.

Table 1: Notations used in the protocol.

| Notations | Descriptions |
|---|---|
| TA | Trusted authority |
| $RSU_i$ | ith roadside unit |
| $Z_q^*$ | A finite field |
| $q$ | A large prime number |
| $ID_{r_i}$ | $RSU_i$'s ID selected by TA during registration |
| $SK_{r_i}$ | Secret key used between RSU and TA |
| $Sv_{r_i}$ | Secret key used to verify legality between RSU and vehicle |
| $V_j$ | $j$th vehicle |
| $s_{v_j}$ | Secret key of $V_j$ |
| $c_{r_i}$ | Random number selected by $RSU_i$ |
| $Tv_j$ | Timestamp selected by $V_j$ during authentication |
| $a_{v_j}$ | Random number selected by $V_j$ during authentication |
| $Tag$ | Denotes the emergency level of the message |
| $r$ | Random number selected by RSU to generate key |
| $GK$ | The group key |
| $M$ | The message sent by vehicle |
| $Q$ | A list that contains legal RSUs' identity |

returns to false. Only then $V_j$ is authenticated by RSU and TA.

(8) Because $RSU_i$ knows $R_{v_1} = H(Sv_{r_i} \| c_{v_1})$ and $b_{v_1}$, it can compute $a_{v_1} = b_{v_1} \oplus H(R_{v_1} \| VID_{v_1})$, and then, $RSU_i$ sends $Q_{r_i} = H(Sv_{r_i})$, $P_{r_i} = H(H(Sv_{r_i}) \| a)$ to $V_j$.

(9) When $V_j$ received these messages from $RSU_i$, it searches its $Q$ list to find whether there exists a $Q_{r_i}$ according to $RSU_i$'s ID and also computes $H(H(Sv_{r_i}) \| a)$ using its own $a$ to verify RSU's identity.

*3.5. Group Key Generation.* The group key generation process is associated with vehicles' movement and time slots. When the region of RSU has no vehicles, the key generation phase is linked with the first vehicle, which is different from the situation that the region already has some vehicles inside.

Assume that there are initially no vehicles in the region of $RSU_i$, when $V_1$ enters into this region and passes the authentication, at any time it wants to send an anonymous message, it should ask for a group key as follows:

(1) $V_1$ sends $O_{v_1} = \{VID_{v_1}, c_{v_1}, b_{v_1}, D_{v_1}, Tv_1\}$ and a $Tag$ to $RSU_i$, where $Tag$ is a sign that indicates the level of information urgency.

(2) Because $RSU_i$ knows $R_{v_1} = H(Sv_{r_i} \| c_{v_1})$, and $b_{v_1}$, it can compute $a_{v_1} = b_{v_1} \oplus H(R_{v_1} \| VID_{v_1})$, then $RSU_i$ choose two random numbers $z_0 \in Z_q^*, G_0 \in Z_q^*$, and compute $GK_1 = a_{v_1} \cdot z_0 \cdot G_0 \bmod q$ as the first new group key. Subsequently, $RSU_i$ chooses a timestamp $T_{r_1}$ and computes $GK_1' = (a_{v_1}^{-1} \cdot G_1 \bmod q) \oplus a_{v_1}$ $Rt_1 = H(GK_1 \| GK_1' \| T_{r_1} \| Q_{r_i})$, and then $RSU_i$ unicasts $GK_1'$, $T_{r_1}$, and $Rt_1$ to $V_1$.

(3) When $V_1$ received these messages from $RSU_i$, it searches its $Q$ list to find a $Q_{r_i}$ which has been already sent by $RSU_i$ during authentication phase according to $RSU_i$'s ID. Then, $V_1$ calculates $GK_1 = (GK_1' \oplus a_{v_1}) \cdot a_{v_1} \bmod q$ and also calculates $Rt_1^* = H(GK_1 \| GK_1' \| T_{r_1} \| Q_{r_i})$, if $Rt_1^* = Rt_1$, and $V_1$ accepts $GK_1$ as the group key.

*3.6. Vehicle Joining Phase.* Assume that in a time slot, a set of vehicles have passed the authentication. There are two situations for group key generation. One is that at the end of the time slot baseline, RSU generates a new group key, unicast it to newly joining vehicles, and multicast it to vehicles, which are already in the group. The other one is that a new vehicle urgently wants to send or request an anonymous message, and it can send a message to RSU to report its desire. RSU then judges the emergency level and dynamically adjusts the time slot; subsequently, unicasting and multicasting phases are the same.

So we will divide this phase into two parts.

*3.6.1. Normal Group Key Generation.* When the time slot came to an end, there are a set of vehicles, which are not so urgent to send anonymous messages that have passed the authentication. The RSU selects a random number $r \in Z_q^*$ and calculates $\overline{GK} = r \cdot GK$ as the updated new key, where $GK$ is the old group key. For each newly joining vehicle, RSU computes $\overline{GK}' = \overline{GK} \oplus H(\overline{T} \| R_{v_j})$, $\overline{Rt}_j = H(R_{v_j} \| \overline{GK} \| \overline{T} \| Q_{r_j})$, and unicasts $\{\overline{GK}', \overline{Rt}_j, \overline{T}\}$ to $V_j$, where $j \in \{1, 2, 3, \ldots, n\}$, $\overline{T}$ is the current timestamp.

When receiving these messages, each $V_j$ calculates $\overline{GK} = H(\overline{T} \| R_j^*) \oplus \overline{GK}'$ to get the updated key $\overline{GK}$ and also calculates $\overline{Rt}_j^* = H(R_{v_j}^* \| \overline{GK} \| \overline{T} \| Q_{r_j})$ according to RSU's ID, if $\overline{Rt}_j = \overline{Rt}_j^*$, and then, $V_j$ accepts the $\overline{GK}$ as the group key. RSU also calculates $\overline{GK}'' = (GK^{-1} \cdot \overline{GK} \bmod q) \oplus GK$ and $\overline{Rt}_j' = H(\overline{GK} \| \overline{GK}'' \| \overline{T} \| Q_{r_j})$ and multicasts $\{\overline{GK}'', \overline{Rt}_j', \overline{T}\}$ to old vehicles. On receiving the message, old vehicles in that group use the old group key $GK$ to compute the new one: $\overline{GK} = (\overline{GK}'' \oplus GK) \cdot GK \bmod q$, and computes $HH(\overline{GK} \| \overline{GK}'' \| \overline{T} \| Q_{r_j})$. If the result equals $\overline{Rt}_j'$, then the $\overline{GK}$ is accepted as the new group key; otherwise, it refused to accept.

*3.6.2. Urgent Conversation Group Key Generation.* Suppose that a vehicle has passed the authentication, however, it finds that the group key has just updated a moment ago through timestamp, and the time slot has just begun. At that moment, it wants to start an urgent group conversation, and it can send a request message to RSU to ask for group key updating.

The urgent vehicle does the following steps:

The vehicle sends $\{M, tag, T, \sigma_j\}$ to RSU, where $M$ is the urgent message, and $tag$ is the level for emergency. $T$ is the timestamp, and $\sigma_j = H(R_j^* \| ID_j)$.

After verification, RSU keeps judging the tag value until it reaches the limited level. Then, the new generation phase will be triggered and time slot will be dynamically adjusted. The dynamic adjustment solution will be discussed in detail in section 4.

Following steps are the same as (1) normal group key generation's all four steps.

*3.7. Vehicle Leaving Phase.* In VANET's communication, vehicles will periodically send beacon messages to inform others that it is still in that region. We suppose that if we did not receive a vehicle's beacon message in 3 cycles, the RSU thinks that node has already left this region. Each cycle is about 2s in general. When RSU thinks that a vehicle has left his region, it triggers a group key updating phase.

The leaving updating phase is as follows:

(1) The RSU selects a random number $n \in Z_q^*$ and computes $\overline{GK}_t = n \cdot \overline{GK}$ as the new group key, where $\overline{GK}$ is the old group key in use. For each vehicle in that group, RSU computes $\overline{GK}_t' = \overline{GK} \oplus H(\overline{Tt} \| R_j)$, $Q_{r_i} = H(Sv_{r_i})$, and $\overline{Rt}_t = H(R_j \| \overline{Tt} \| \overline{GK}_t \| Q_{r_i})$. Then, RSU sends $\{\overline{Rt}_t, \overline{GK}_t', \overline{Tt}\}$ to all vehicles.

(2) After receiving the message, each $V_j$ searches its $Q$ list to find whether there exists a $Q_{r_i}$ according to $RSU_i$'s ID. Then, $V_j$ computes $\overline{GK}_t = \overline{GK}_t' \oplus H(\overline{Tt} \| R_j^*)$ to get $\overline{GK}_t$ and computes $H(R_j^* \| \overline{Tt} \| \overline{GK}_t \| Q_{r_i})$, if the result matches $\overline{Rt}_t$, and then, $V_j$ accepts $\overline{GK}_t$ as the updated group key.

## 4. Dynamic Time Slot Adjustment

In this section, we discuss the time slot adjustment in detail.

In many works of literature, to reduce communication costs, we usually aggregate a set of vehicles' authentication processes, which is known as "batch" authentication. Many of them use mathematical methods, which may lead to a large number of calculations. A more straightforward approach is to use time slot, in which we split time into intervals. RSU generates different keys in each interval. This introduces a delay, maybe half of the slot time generally, but allows to reduce the number of rekeying events. Note that our slot's length is different according to different situations. So, this will achieve efficiency and cost a balance.

Suppose we are in a large density of vehicle environment, if the frequency of key update is very fast, for example, we encrypt the message with key A and send it out. After the destination receives it, the key A may be expired, so a lot of time is spent on group key verification phase, which is not worth the candle.

Next, we will discuss the dynamic time slot strategy from the aspect of the vehicle initiating the request and the departure of the vehicle.

*4.1. LSTM Model for Traffic Trajectory Prediction.* Our model consists of three steps: data unfolding, similarity measurements, and ensemble quality prediction. For each RSU, the historical dataset is collected by recent traffic flow, which includes distance, angle, and trend in its area. Considering the storage capacity of the RSU, the dataset can be stored on the cloud within the stipulated time. After implementing data normalization of the historical dataset $X_H$, RSU also needs to collect the real-time query sample as $\{x_q, x_{q+1}, \ldots, x_{q+n-1}\}$. With the dataset and the real-time sample, we tried to calculate different similarity measurements and extract the modeling trajectories for each strategy on the cloud. After modeling trajectories have been proposed, we further need to construct online local soft-sensing models as $y = f_b(x)$ for different strategies.

From the above models, the predicted results of different models can be further proposed:

$$\{y_{q,b}, y_{q+1,b}, \ldots, y_{q+n-1,b}\}. \quad (1)$$

Then, we use cross-validation strategy to determine the weight $\eta_b$, which will be used in next step. With all the above

results, we can get the weighted sum of local models as follows:

$$\{y_q, y_{q+1}, \ldots, y_{q+n-1}\}. \quad (2)$$

*4.2. Joining Phase's Urgent Request.* In general situations, we assume that the time slot baseline is 10 seconds. If a vehicle has already passed the authentication, and it has a desire to send anonymous messages at some time when the next time slot has not come yet, it sends a request to RSU in the following format:

$$R = \{M.Tag, T_{req}\}. \quad (3)$$

$M$ is the message the vehicle wants to broadcast, *Tag* is the symbol that denotes the emergency level, and $T_{req}$ is the current time.

We divided the message emergency level into five degrees, represented by the value of the *Tag,* 5 is the highest level, and 1 is the lowest level. The higher the level, the more urgent the message is. The critical message is, for example, someone wants to report road condition, or wants to know road condition ahead of him. Another message like requesting multimedia resources is not that urgent. Each Tag's value ranges from a predefined time slot $t'$.

However, it should be noted that the value of the *Tag* is not determined by the user. Trusted authorities should have an agreement in advance on the weight of various messages. Five levels are sufficient to cover most message levels.

In addition, we limit the number of the vehicle's requests to three times in one RSU region, to avoid malicious nodes sending too many requests.

On receiving $R$, RSU first computes $\Delta T = T_{next} - T_{req}$, and the $T_{next}$ denotes the time of next key update, which is no more than a certain time slot $t$. So $\Delta T$ ranges from the beginning of the time slot to the end of it.

Then, RSU computes $W = \Delta T + Tag$, so $W$ is associated with both $t$ and $t'$. Then, we use the weight $\eta_b$ defined by LSTM to set the threshold value to trigger the key updating phase.

The RSU performs the following steps when receiving request messages:

(i) RSU has a request queue *Queue*, when the request message arrives, and RSU appends $\Delta T$ to $R$ and puts them into *Queue.*

(ii) RSU then gets $W$ and compares $W$ with $\eta_b$, and if there is still under $\eta_b$, RSU then waits for other requests. After each message arrives, $W$ will be accumulated, so finally we get an accumulated $\Sigma W$, if at the end of the time slot, $\Sigma W$ still failed to reach $\eta_b$, and then, RSU starts a new updating phase.

(iii) If the $\Sigma W$ reaches $\eta_b$, RSU starts updating phase at once.

*4.3. Leaving Phase's Updating Strategy.* Because we cannot exactly judge a vehicle's leaving time, we think that if we have not received vehicle's beacon broadcast message within three cycles, the vehicle has already left this region. At that time, a new group key updating phase will be triggered. At this part, we do not use the 10-second strategy, because that we want to forbid a vehicle from keeping a valid group key when it is not in that region. This is more influential than entering the area but not yet having a group key. The updating steps are the same as Part III, 3.7 the vehicle leaving phase. After the group key's updating, a new time slot will begin.

## 5. Analysis and Comparison

We divide this section into three parts, the first part is the security analysis of the authentication, the second part is the computational cost comparison of our authentication process with another process, and the third part is the group key request's average waiting delay. We compare the execution time of the cryptographic operations using JPBC [26], which is a famous cryptography library and has been widely used to implement cryptographic operations in many environments. Our hardware platform consists of an Intel(R) Core(TM) i5-6600K processor with 3.50 GHz clock frequency, 16 gigabytes memory, and runs Windows 10 operating system. The execution times of the above cryptographic operations are listed in Table 2.

*5.1. Security Analysis*

(1) Authentication*:* When authenticating with RSU, the vehicle $V_j$ should use its own $ID_{v_j}$ to compute $VID_{v_j}$. Then, $RSU_i$ verifies $ID_{v_j}$'s legality with help of TA. The communication between TA and $RSU_i$ is considered to be safe. In this way, the legality of the vehicle can be verified because the ID of the vehicle has only been confirmed by TA during registration and is only held by a legitimate vehicle.

Then, $RSU_i$ computes $a_{v_1} = b_{v_1} \oplus H(R_{v_1} || VID_{v_1})$ to get $a_{v_j}$ and sends $Q_{r_i} = H(Sv_{r_i})$, $P_{r_i} = H(H(Sv_{r_i}) || a)$ to $V_j$, Because TA has already sent a list $Q$ to $V_j$, in this way $V_j$ can authenticate RSU at the same time. After authentication, in group key generation phase, $V_j$ can use this $Q_{r_i}$ to validate RSU's legality.

(2) Traceability: If an abnormality is found, the RSU will report the abnormal vehicle's ID to the TA using the secret $SK_{r_i}$ between TA and him, and then, TA will take sanctions.

For the vehicles that do not correctly report their position or other information, the RSU first will identify the exact position of the vehicle and report the vehicles' ID to the nearby RSU by its trajectory. For the vehicles that do not have a consistent trajectory, RSU cannot search within its area but to report the ID of the vehicle to TA and other RSU nearby.

(3) Replay attack: The timestamp $\overline{T}$ is used to protect the replay attack and keep the freshness of authentication, RSU will also send $\overline{Rt}_j = H(R_{v_j} || \overline{GK} || \overline{T} || Q_{r_j})$ to vehicle, and then, the vehicle will detect replay attack after verifying $\overline{Rt}_j$.

Table 2: Running times of different operations.

| Operation | Execution time (ms) |
|---|---|
| $T_{pair}$: bilinear pairing | 4.2 |
| $T_{mtp}$: map to point hash function | 4.4 |
| $T_{mul}$: scaler multiplication | 0.4 |
| $T_h$: general hash operation | 0.0001 |
| $T_L$: operation time on LSTM | 0.737 |
| $T_{ecc-mul}$: execution time for calculating the elliptic curve point multiplication | 0.442 |
| $T_{ecc-a\ dd}$: execution time for elliptic curve point addition | 0.0018 |
| $T_{pair-a\ dd}$: execution time for point addition to bilinear pairing | 0.0071 |
| $T_{\exp}$: execution time for exponential operation | 0.6 |

Table 3: Execution time of different schemes.

| Scheme | Operation | Execution time (ms) |
|---|---|---|
| EAAP | $2T_{\text{pair}} + 5T_{\exp}$ | 11.4 |
| J. Zhang et al. | $3T_{pair} + 8T_{mul} + T_{mtp} + 3T_{pair-a\ dd} + 7T_h$ | 20.1 |
| M. Bayat et al. | $6T_{mul} + T_{pair} + 2T_{mtp} + T_{pair-a\ dd} + 3T_h$ | 15.4 |
| Lo and Tsai | $5T_{ecc-mul} + 4T_h + 2T_{ecc-a\ dd}$ | 15.4 |
| Our authentication scheme | $10T_h + T_L$ | 0.001 |

(4) Backward and forward secrecy: The new group key generation is associated with the old group key and a random number, such as $\overline{GK} = r \cdot GK$, the newly joined vehicle cannot compute the old key because they do not know the old group key and that random number, and therefore, backward secrecy is guaranteed. A leaving vehicle, even if he keeps the old group key, will not know the random number, so forward secrecy is guaranteed.

(5) Impersonation attack: If a malicious node wants to impersonate a vehicle, it cannot pass the authentication without knowing the legal ID of the vehicle, and this ID is securely delivered to the vehicle during registration with TA. If a third party wants to impersonate RSU, it should have the legal $Sv_{r_i}$ to compute $H(H(Sv_{r_i})||a)$ to obtain the vehicle's trust. Thus, we can resist the impersonation attack.

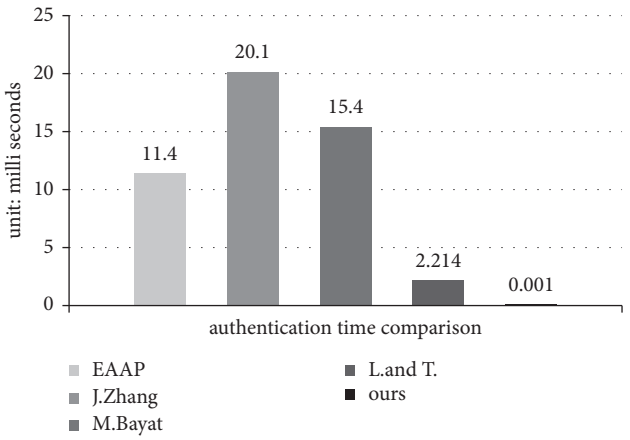


Figure 3: Authentication time comparison.

*5.2. Authentication Process Computation Comparison.* In this part, we compare our scheme with EAAP [27], Zhang et al. [28], Lo and Tsai [13], and M. Bayat [29] schemes. First, we have some notations for running times of operations:

Next, we give the different execution steps for each scheme and compare their execution time with ours. The total time includes authentication message generation time and authentication message verification time but neglects the nondominant operations.

In EAAP, J. Zhang, and M. Bayat et al.'s articles, they use at least one pairing operation and one map to point hash function, which is known as time-consuming operations. Complex operations bring higher security but result in more computation costs. Considering that OBU has limited computing power, in some scenarios, we can reduce communication pressure by using some secure and fast operations like hashing.

From Table 3 and Figure 3, we can clearly see that our authentication scheme has the lowest computation cost, for which we do not have those complicated operations.

*5.3. Batch Group Key Request Waiting Delay Analysis.* In this part, we compare the average delay time for newly joining vehicles' group key request at different vehicle joining densities. Suppose the average delay time is significantly less than the cycle time, the vehicle needs to request some information. In this case, we think the dynamic adjustment strategy is considered feasible. Besides, we think that the group key update caused by the vehicle leaving event will not affect the user's experience, so no test analysis will be conducted.

In the experiment, we let the vehicles randomly enter the area at a frequency of 10, 20, 30, and 50 vehicles every 10 seconds. The protocol will dynamically adjust the slot time based on the baseline slot. The algorithm pseudo-code used in the experiment is as follows. Next, we will explain it in detail.

In a baseline time slot, we assume that vehicles randomly enter the area at a different frequency. When some vehicles need to send requests or share information, they initiate

**Input**: number of nodes $n$, average request cycle time tp.
**Output**: average request waiting delay

```
(1) request = = false;
(2) while(int i ≤ n)
(3) { if(request)
(4)    { ΔT = Tnext − Treq;
(5)      W = ΔT + Tag;
(6)      queue.add(W);
(7) for(int m = 0; m < queue.size(); m++)
(8)      { int sum+ = queue.get(m)
(9)   if(sum ≥ 7){
(10)          then trigger a new update;
(11)          Tnew = current time; }
(12)               break;
(13)     }
(14)   }
(15) }
(16) for(int m = 1; m<= Δn; m++)
(17)     { Δt_m = Tnew -Treq_m;
(18)       sum' + = Δt_m;
(19)     }
(20) ΔT = sum'/Δn;
(21) return ΔT;
```

Algorithm 1: Average waiting time experiment process.

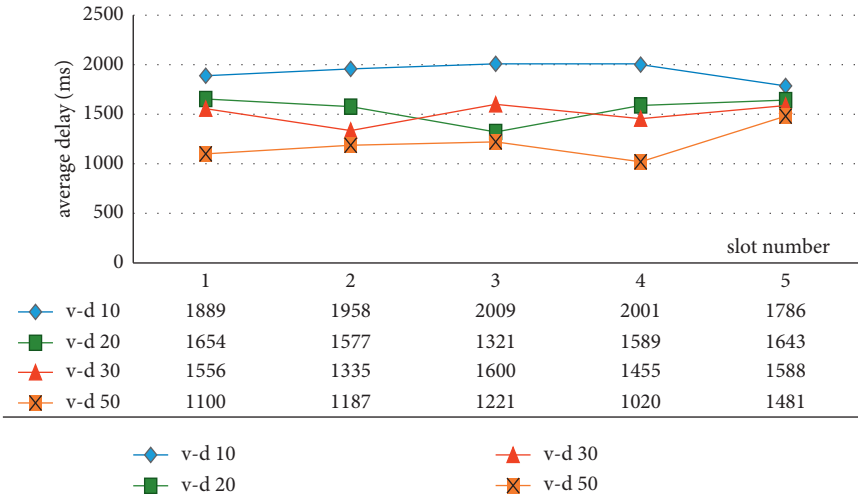


Figure 4: New vehicles' average waiting time.

requests to the RSU. Then, RSU judges the weights according to the method described in section 3, part A. Whenever a request arrives, the RSU puts its weight W value into the entering queue. When the sum of W reaches a threshold value, then RSU triggers a new group key update.

*Tnext* denotes the next slot starting time, *Treq* denotes the moment when the vehicle sends a request. $\Delta n$ denotes the number of the vehicles, which send request messages. When the new updating phase has been triggered, we calculate the waiting time $\Delta t_m$ for every newly joining car who sends request to RSU and add them up. At last, we get the average waiting time in this time slot.

The next figure shows the average waiting time at different vehicle entering rates.

In Figure 4, v-d denotes vehicle density. From the experimental results, we can see that as the vehicle density increases, the key update time becomes faster, and the average waiting time of the new requesting vehicle becomes shorter. When the enter frequency is 50 vehicles in 10 seconds, the average waiting time is less than 1.5s, which is generally less than the ordinary communication frequency. In this study, we just consider nonperiod messages, which are usually longer than the 1- to 2-second cycle time of the beacon message. So, we can say that the protocol meets the requirement.

## 6. Conclusions

In this study, we first describe a lightweight authentication protocol, which is less complex and has a lower computation overhead. This protocol can achieve the same security as traditional protocols. RSU plays the role of group manager and authenticates vehicles with the help of the TA. We also have a simple group key generation and authentication process, which overcomes lots of problems in group authentication. Second, to reduce the communication cost and the frequency of the group key updating phase, we design a dynamic batch-based group key generation method that can fit the need of urgent communication and reduce communication overhead. In the urban area, if the entry and exit of each vehicle cause a group key update, the intensive vehicle makes the update of the group key very frequent, so we make a decision whether to immediately update the group key based on the urgency of the message and the time difference from the new slot. To achieve a balance in this way, we introduce an LSTM method to predict the trajectory of each vehicle to divide the message into different groups and use batch key management. The experiment results show that our authentication protocol has a smaller computation overhead, and under the dynamic time slot adjustment strategy, the vehicle's average waiting time is short, within a tolerable communication cycle.

## Data Availability

The data used to support the findings of this study are available from the authors upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Natural Science Foundation of China project (nos. 61772385 and 61572370);

## References

[1] S. Hammoudi, Z. Aliouat, and S. Harous, "Challenges and research directions for Internet of things," *Telecommunication Systems*, vol. 67, no. 2, pp. 367–385, 2018.

[2] F. Qu, Z. Wu, F. Wang, and W. Cho, "A security and privacy review of VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2958–2996, 2015.

[3] B. G. Premasudha, V. Ravi Ram, and J. Miller, "A review of security threats, solutions and trust management in vanets," *International journal of Next-Generation computing*, vol. 7, no. 1, pp. 38–57, 2016.

[4] S. S. Manvi and S. Tangade, "A survey on authentication schemes in VANETs for secured communication," *Vehicular Communications*, vol. 9, pp. 19–30, 2017.

[5] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1227–1239, 2010.

[6] L. Delgrossi and T. Zhang, *Vehicle Safety Communications: Protocols, Security, and Privacy*, WILEY, 1st edn edition, Sep. 2012.

[7] D. Chaum and E. van Heyst, "Group signatures," *Advances in Cryptology - EUROCRYPT '91*, vol. 547, pp. 257–265, 1991.

[8] A. Wasef and X. Shen, "Efficient group signature scheme supporting batch verification for securing vehicular networks," in *Proceedings of the IEEE International Conference on Communications(ICC'2010)*, Cape Town, South Africa, May 2010.

[9] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[10] 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.

[11] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proceedings of the IEEE INFOCOM*, pp. 816–824, Phoenix, AZ, USA, April 2008.

[12] C. Zhang, P.-H. Ho, and J. Tapolcai, "On batch verification with group testing for vehicular communications," *Wireless Networks*, vol. 17, no. 8, pp. 1815–1865, 2011.

[13] N.-W. Lo and J.-L. Tsai, "An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 13–19, 2016.

[14] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," *Advances in Cryptology - CRYPTO 2004*, vol. 3152, pp. 41–55, 2004.

[15] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: a secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, 2007.

[16] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: efficient conditional privacy preservation protocol for secure vehicular communications," *Proc. INFOCOM*, vol. 2008, pp. 1229–1237, 2008.

[17] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in VANET," in *Proceedings of the 4th ACM international workshop on Vehicular ad hoc networks*, pp. 19–28, Quebec, Montreal, Canada, September 2007.

[18] Y. Hao, Y. Cheng, C. Zhou, and W. Song, "A distributed key management framework with cooperative message authentication in vanets," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 616–629, 2011.

[19] P. Vijayakumar, M. Azees, A. Kannan, and L. J. Deborah, "Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1015–1028, 2016.

[20] L. Zhang, Z. Huang, W. Liu, Z. Guo, and Z. Zhang, "Weather radar echo prediction method based on convolution neural network and Long Short-Term memory networks for sustainable e-agriculture," *Journal of Cleaner Production*, vol. 298, no. 8, p. 126776, 2021.

[21] L. Zhang, C. Xu, Y. Gao, Y. Han, X. Du, and Z. Tian, "Improved Dota2 lineup recommendation model based on a bidirectional LSTM," *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 712–720, 2020.

[22] Z. Gu, T. Shen, Y. Wang, and F. C. M. Lau, "Efficient rendezvous for heterogeneous interference in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 91–105, 2020.

[23] Z. Gu, W. Hu, C. Zhang, H. Lu, L. Yin, and L. Wang, "Gradient shielding: Towards understanding vulnerability of deep neural networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 921–932, 2021.

[24] Z. Gu, Y. Wang, T. Shen, and F. C. M. Lau, "On heterogeneous sensing capability for distributed rendezvous in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 11, pp. 3211–3226, 2021.

[25] X. Shen, J. Yang, L. Zhang, and G. Yang, "An interactive role learning and discovery model for multi-department RBAC building based on attribute exploration," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 1373–1382, 2022.

[26] JPBC Library, [online] Available: http://libeccio.di.unisa.it/projects/jpbc/, 2012.

[27] M. Azees, P. Vijayakumar, and L. J. Deboarh, "EAAP: efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc net works[J]," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 10, pp. 1–10, 2017.

[28] J. Zhang, M. Xu, and L. Liu, "On the security of a secure batch verification with group testing for VANET," *International Journal on Network Security*, vol. 16, no. 5, pp. 355–362, 2014.

[29] M. Bayat, M. Barmshoory, M. Rahimi, and M. R. Aref, "A secure authentication scheme for VANETs with batch verification," *Wireless Networks*, vol. 21, no. 5, pp. 1733–1743, 2014.