Delegating Authentication to Edge: A Decentralized Authentication Architecture for Vehicular Networks

Anjia Yang, Member, IEEE, Jian Weng, Member, IEEE, Kan Yang, Member, IEEE, Cheng Huang, Member, IEEE, and Xuemin Shen, Fellow, IEEE

Abstract—Secure and efficient access authentication is one of the most important security requirements for vehicular networks, but it is difficult to fulfill due to potential security attacks and long authentication delay caused by high vehicle mobility, etc. Most of the existing authentication protocols, either do not consider attacks like single point of failure or do not focus on reducing authentication delay. To address these issues, we introduce an edge-assisted decentralized authentication (EADA) architecture, which provides secure and more communication-efficient authentication by enabling an authentication server to delegate its authentication capability to distributed edge nodes (ENs) such as roadside units (RSUs) and base stations (BSs). Under the architecture, we propose a threshold mutual authentication protocol that supports fast handover, which involves two scenarios, Auth-I and Auth-II. Auth-I only happens once when a vehicle tries to access the network for the first time, while Auth-II happens when a vehicle seamlessly roams between two ENs, i.e., handover. Specifically, for Auth-I, each vehicle can be cooperatively authenticated by $t$ out of $n$ ENs with identity-based signature techniques to obtain an authentication token and the involved ENs can be efficiently authenticated in a batch by the vehicle. For Auth-II, the vehicle can utilize the token as its private credential to achieve fast handover based on identity-based signature without interacting with multiple ENs, which further reduces the authentication delay significantly. In addition, we design a flexible method to support dynamic joining and leaving of ENs without the assistance of a trusted center. We demonstrate that the proposed protocol is secure and efficient through security analysis and performance evaluation.

Index Terms—Vehicular networks, edge, mutual authentication, threshold signature.

I. INTRODUCTION

The advanced sensing and communication technologies have been integrated into vehicles to make them more intelligent. Through dedicated short-range communication (DSRC) [1] or LTE V2X technology [2], [3], vehicles can interact with other vehicles or infrastructures to share and exchange information, which forms vehicular networks that have been well recognized as one of the potential components to achieve intelligent transportation systems (ITS). A typical vehicular network is composed of vehicles and infrastructures named roadside units (RSU) that are sparsely deployed over roads. With vehicular networks, RSUs can broadcast critical information such as traffic accidents and road conditions to surrounding vehicles which can improve the driving safety and the energy efficiency of road travel, while in the meantime vehicles can request services such as downloading navigation maps and automated valet parking upon driving through the communication area of a certain RSU. For example, a project in Europe called SCOOP [4] has been built for deploying cooperative intelligent transport systems based on vehicular networks in order to improve road safety. Especially, with the interconnection and accessibility problems of vehicular networks having been firstly addressed by Cheng et al. [5], it becomes more practical to deploy vehicular networks for ITS.

Similar to other wireless networks, in vehicular networks, data is transmitted directly over the air, which means the communication channel is vulnerable to adversaries and various security threats like impersonation, replay and man-in-the-middle attacks [6]–[8] could be launched. Security is an essential factor to ITS, since illegally accessing or tampering data may lead to severe consequences and threaten national security. Specifically, to ensure that only authorized vehicles...
can request services (e.g., wireless network access), it is critical to verify the authenticity of vehicles. Due to the high mobility of vehicles and the limited communication range of RSUs (e.g., 300-500 meters) [9], a moving vehicle may stay within a specific RSU only for a short time and may need to reconnect to another RSU frequently, which means the vehicle is supposed to be authenticated. However, frequent authentication may severely degrade the quality of networking services for vehicles. As a consequence, designing an effective authentication scheme for vehicular networks confronts more challenges.

A common authentication method as shown in Figure 1(a) is to employ an authentication server (AS) (e.g., an AS is employed by the Department of Motor Vehicles in U.S.) to store some “secret information” about the vehicle during the registration, which can further be exploited to authenticate the vehicle by the AS later on. When a vehicle requests for some networking service, it needs to authenticate itself to AS through an RSU. However, when the number of vehicles becomes large, the burden of authenticating all the vehicles at the same time is extremely large for the single AS which turns out to be a bottleneck of the system and also leads to the single point of failure problem. To avoid these issues, an improved authentication architecture is shown as in Figure 1(b), which employs multiple ASs to authenticate vehicles. A proper protocol under this architecture can indeed amortize the authentication burden and mitigate the single point of failure issue, but it also inevitably poses delay because of the interaction between the RSU and the remote ASs. To avoid the authentication delay, some researchers [10], [11] proposed to delegate the authentication functionality to fog nodes (e.g., RSUs) which are close to vehicles compared with remote authentication servers. This kind of architecture as shown in Figure 1(c) indeed reduces the authentication latency, but it does not consider the fog node compromise attack. Once a fog node is controlled, the security of the system is broken.

To cope with all the above challenging issues, we propose a new edge-assisted decentralized authentication architecture (EADA) for vehicular networks. The architecture includes three types of parties, i.e., a registration server (RS), edge nodes (EN) (e.g., RSUs and BSs), and vehicles. Specifically, RS is in charge of registration and revocation for vehicles and ENs, while ENs cooperatively authenticate vehicles and each vehicle can authenticate the ENs as well. Within this architecture, a vehicle requesting services can be authenticated by ENs directly without the assistance of the remote RS. This can reduce the computation and communication overhead as well as the authentication delay compared with the traditional authentication architecture. Note that RS only takes care of the registration and management (e.g., update and revocation of a registered vehicle) and thus can be offline during the authentication phase, which makes it more difficult for attackers to compromise RS. Furthermore, to resist the single point of failure problem, we propose to employ threshold cryptosystems [12] where multiple ENs collaboratively authenticate a vehicle. In this case, even compromising a limited number of ENs will not disrupt the system. In summary, the contribution of this article is briefly summarized as follows.

- We propose an edge-assisted decentralized authentication architecture (EADA) for vehicular networks, where the authentication capability is delegated to distributed edge nodes (e.g., RSU and BS). This architecture can assist in providing more communication-efficient authentication since the ENs are introduced as middlewares between vehicles and the registration server.
- Under the authentication architecture, we propose a scalable threshold mutual authentication protocol supporting fast handover, which is divided into two cases: Auth-I and Auth-II. Auth-I only happens once when a vehicle tries to access the network for the first time, while Auth-II happens when a vehicle seamlessly roams between two ENs. Specifically, for Auth-I, each vehicle can be cooperatively authenticated by $t$ out of $n$ ENs with identity-based signature techniques to obtain an authentication token and the involved ENs can be efficiently authenticated in a batch by the vehicle. For Auth-II, the vehicle can pre-compute its authentication credentials based on the token that is regarded as its secret to achieve fast handover by its nearest EN based on identity-based signature without interacting with multiple ENs, which further reduces the authentication delay significantly.
- We propose a flexible method to support dynamic joining and leaving of ENs without the involvement of a trusted center.

Furthermore, we provide a comprehensive security analysis and evaluate the performance of the proposed protocol. We also conduct a series of comparison experiments and the experimental results show that our protocol is efficient in terms of both computation complexity and communication latency.
The remainder of this article is organized as follows. In Section II, we revisit existing works that are related to ours. Then we introduce preliminaries to be used in design of the proposed scheme in Section III. In the next section, system model, security model and design goals are defined. The proposed protocol is elaborated in Section V, followed by the security analysis and performance evaluation of the proposed protocol. Finally, we make a conclusion of this article in Section VIII.

II. RELATED WORK

To provide secure communication for vehicular networks, Raya and Hubaux [6] proposed to utilize PKI as a building block for vehicle authentication where the certificates of vehicles were transmitted along with the credentials. After that, many different authentication protocols [5], [13]–[30] have been presented with focusing on distinct functionalities. For example, Jiang et al. [14] had concerns of the efficiency of vehicular authentication protocols that adopt certificate revocation lists (CRL), since it is necessary to query the CRL and the certificate of a vehicle in order to verify its authenticity, both of which are time consuming. Instead of the CRL method, they proposed to check the revocation status of a vehicle by a fast HMAC function which can reduce the authentication delay significantly. He et al. [16] and Azees et al. [17] dealt with the efficiency of conditional privacy-preserving authentication protocols. Zhang et al. [19] proposed a distributed aggregate authentication scheme for vehicular networks. They considered RSUs as lower-level trusted authorities which are enrolled by a root trusted authority. Vehicles can register to any of these RSUs and achieve the authentication based on their identities and public keys of the RSUs. Hao et al. [31] and Jo et al. [18] proposed cooperative message authentication where multiple vehicles collaboratively verify a message in order to reduce the authentication cost on individual vehicles. Zhang et al. [15] proposed a novel Chinese remainder theorem (CRT)-based conditional privacy-preserving authentication protocol, which only requires realistic TPDs instead of ideal TPD. This will greatly promote the deployment and application of the authentication protocol in VANETs. These message authentication protocols focus on guaranteeing the authenticity of the party who sends the messages and cannot be directly adapted to user authentication in service-oriented systems.

Huang et al. [10] introduced the concept of vehicular fog computing that integrates vehicular networks with fog computing techniques. Specifically, RSUs are regarded as fog nodes which can help process requests from vehicles. Thakur and Malekian [32] employed fog computing based connected vehicles to handle vehicular congestion. Yao et al. [11] proposed a blockchain-based anonymous authentication for distributed vehicular fog services. In their protocol, only the fog node closest to the vehicle is required to authenticate vehicles, after which the fog node broadcasts the authentication result to other fog nodes and record the results to blockchain. Therefore, their protocol did not consider fog node compromise attack. Once a fog node is controlled or compromised, the reported result to the blockchain could be misleading. Cui et al. [33] introduced edge computing into authentication for vehicular networks, where parts of vehicles are considered as the edge nodes to assist RSUs to authenticate other nearby vehicles. After that, RSUs broadcast the authentication results for the messages through a cuckoo filter. The legitimacy of these messages is thus able to be quickly determined by querying the filter later. Ma et al. [13] proposed a secure authenticated key agreement protocol for fog-based vehicular networks which does not utilize bilinear pairing and thus becomes very computationally efficient. Nevertheless, they employ the remote cloud server (i.e., the authentication server) instead of fog nodes to authenticate vehicles and thus cannot avoid the delay between the cloud server and the fog nodes.

Shao et al. [34] designed a threshold authentication protocol for vehicular networks, in which a message should be accepted only after it has been verified by a threshold number of vehicles. Nevertheless, their work considered message authentication instead of user authentication and therefore cannot be applied to solve the problems defined in this article. Malik et al. [35] proposed to authenticate vehicles assisted with blockchain technology, which can avoid the single-point-of-failure problem. However, the authentication delay includes the time when RSUs query certificates from the blockchain which is not discussed in that paper. Park et al. [36] proposed an efficient RSU-based distributed group key management for vehicular networks, where the key distribution center takes charge of vehicles membership updates and each RSU is responsible for generating and managing keys for its serving vehicles. Different from their scenario, we consider that the fog nodes (e.g., RSUs) could be compromised by attackers and thus do not manage keys for vehicles in this article.

Except for normal authentication, handover is introduced to enable mobile devices to seamlessly and securely roam among multiple access points in mobile wireless networks. Likewise, in vehicular networks, when a vehicle moves to a new access point (the coverage of a new RSU), the handover authentication should allow the new access point to authenticate the vehicle as well. There have been a lot of handover authentication protocols designed in the literature [37]–[39]. As mentioned in the introduction part, all these authentication protocols require active participation of the authentication server, which will cause delay due to the interaction between the access point and the remote authentication server. As a contrast, our proposed architecture delegates the authentication to local edge nodes which thus eliminates the handoff delay across multiple RSUs.

III. PRELIMINARY

A. Bilinear Pairing

Let G and GT be an additive and multiplicative cyclic group respectively, both with order q (q is a large prime). Practically, G may be implemented utilizing a group of points on some elliptic curve over a finite field while GT may be implemented utilizing a multiplicative subgroup of a finite field. A map e : G × G → GT is said to be an admissible bilinear pairing if it satisfies the following properties:

1) Bilinear: e(aP, bQ) = e(P, bQ)a = e(aP, Q)b = e(P, Q)ab for any P, Q ∈ G and any a, b ∈ Z∗

G.
2) Non-degenerate: there exists \( P \in G \), so that \( e(P, P) \neq 1 \) (i.e., \( e(P, P) \) is a generator of \( G_T \)).
3) Computable: there exists an efficient algorithm to compute \( e(P, Q) \) for any \( P, Q \in G \).

B. Threshold Cryptography

First introduced by Shamir [40], a \((t, n)\)-threshold scheme allows a dealer owning a secret \( s \in Z_q^* \) to distribute its secret to \( n \) parties, so that any \( t \) or more out of the \( n \) shareholders are able to reconstruct \( s \) while less than \( t \) parties cannot reveal any information about \( s \) even they collude. This can be done with the following steps. The dealer selects a polynomial \( f(x) \) over \( Z_q \) of degree \( t - 1 \) so that \( s = f(0) \) and sends each party an evaluation of \( f(x) \) on points \( i = 1, \cdots, n \) (i.e., \( (f(i)) \)). Then given any subset of \( t \) pairs from the set of \( \{(i, f(i))\} \), without loss of generality saying \( (i_1, f(i_1)), \cdots, (i_t, f(i_t)) \), the secret \( s \) can be recovered by \( s = \sum_{j=1}^{t} f(i_j) \prod_{i \neq j} \frac{1}{i_i - i_j} \). Based on Shamir’s work, various threshold signature schemes [41]–[45] have been constructed. A typical application is to use threshold signature for authentication purpose to prevent single point of failure [34], [46]–[48], which is also the usage purpose in this article.

C. Mathematical Assumptions

The security of the proposed protocol is based on the intractability of some mathematical assumptions which are described as follows.

1) Discrete Logarithm (DL) Assumption: Given a generator \( P \in G \) and a random chosen element \( X \in G \), there is no probabilistic polynomial adversary who can find an integer \( a \in Z_q^* \) so that \( X = aP \) with non-negligible probability.
2) Computational Diffie-Hellman (CDH) Assumption: Given a tuple of \((P, aP, bP) \in G^3 \) with randomly selected \( a, b \) from \( Z_q^* \), there is no probabilistic polynomial adversary who can compute \( abP \in G \) with non-negligible probability.
3) Bilinear Diffie-Hellman (BDH) Assumption: Given a tuple of \((P, aP, bP, cP) \in G^4 \) with randomly selected \( a, b, c \) from \( Z_q \), there is no probabilistic polynomial adversary who can compute \( e(P, P)^{abc} \in G_T \) with non-negligible probability.

IV. SYSTEM MODEL, THREAT MODEL, AND DESIGN GOALS

A. System Model

As shown in Figure 2, we consider an edge-assisted vehicular network which comprises a registration server, a number of edge nodes (acting as authentication servers) and vehicles. Edge nodes are connected with diverse service providers through secure high-bandwidth backbone networks and thus can provide various services to legitimate vehicles. The detail of each entity is elaborated as follows:

- Registration Server (RS): it is a trusted party that initializes the whole system and takes care of the registration of vehicles and ENs. Upon registration, RS generates secret keys for vehicles and ENs that would be used in authentication phase. It is also responsible for notifying the update information of vehicles such as joining or revocation to ENs. Once the registration is done, RS could be offline since the authentication procedure can be finished by vehicles and ENs themselves.
- Vehicle: a vehicle is an end-user that wants to gain services from vehicular networks. Each vehicle is installed with on-board units that are used to communicate with ENs. In general, vehicles have adequate computational resource and storage capability to perform public cryptography operations, which is practical considering the powerful CPU that modern vehicles have.
- Edge Node (EN): an EN (e.g., RSU or a BS) is located on the roadside and can interact with vehicles directly. The authentication capability is distributed to a set of ENs from traditional authentication servers. Namely, whenever an EN receives a request message from vehicles under its coverage, it will authenticate the vehicle, collaboratively with other ENs if necessary. The computational and storage capabilities of ENs should be far greater than vehicles.

With the above entities, we briefly introduce how the system works from scratch as shown in Figure 2. First, all the vehicles and ENs register to RS with their identities to obtain their individual private keys. Then when a vehicle \( V \) enters the communication range of an EN, it can request for expected services through the EN. Before obtaining the access permission, \( V \) should be authenticated. If it is \( V \)’s first request without a valid token, \( V \) will be authenticated by a set of ENs together and meanwhile these ENs could also be authenticated by \( V \). If all parties pass the authentication, these ENs will collaboratively generate a valid token with an expiry time to \( V \). If \( V \) has already a valid token, it can be fast authenticated by its nearest EN based on this token.

B. Threat Model

The registration server RS is assumed to be a trusted party and cannot be compromised. Thus, it will honestly follow the protocol to provide registration and membership updates services. Vehicles are considered to be malicious in sense that they may try to forge valid tokens to pass the authentication. Similarly, ENs could be malicious or compromised by attackers. However, to ensure the system security, we assume that attackers can only compromise a limited number of ENs,
where the value of the number is defined according to the system parameter. The communication channel between vehicles and ENs is vulnerable to attackers. Namely, attackers can eavesdrop, inject, send and even modify messages transmitted on the channel. The communication channel among different ENs is assumed to be secure against the adversary by utilizing existing techniques like Transport Layer Security protocol (TLS). With the above assumption, we consider a probabilistic polynomial adversary which may launch the following attacks:

1) **Replay attack**: an attacker eavesdrops and records previous transaction transcripts between vehicles and ENs and then disguises as a legitimate vehicle using the recorded transcripts in order to pass a new authentication with the ENs.

2) **Impersonation attack**: an attacker impersonates a legitimate vehicle to convince the ENs to accept it as the legitimate vehicle so that it can obtain the access permission.

3) **EN compromise attack**: an attacker could compromise a set of ENs, aiming to break down the authentication mechanism so that it can abuse the services. Namely, it can grant access permissions to any vehicles no matter they are legal or not.

4) **Wasting network resources attack**: an attacker can flood a large number of requests to a specific EN which will forward these requests to other ENs for collaborative authentication, with the purpose to waste the network resources.

5) **Token forgery attack**: an attacker may forge an authentication token without secret keys in order to pass the authentication.

C. Design Goals

Based on the system model and security threats elaborated above, we define a series of design goals for the proposed threshold authentication protocol in terms of three aspects: efficiency, security, and scalability, elaborated as the following.

1) **Efficiency**: The communication delay caused by authenticating vehicles should be reduced as much as possible. The protocol should also be efficient in terms of computation complexity.

2) **Security**: The proposed protocol should satisfy the following security requirements.
   - Mutual Authentication: It should achieve mutual authentication between vehicles and ENs. Namely, vehicles can authenticate ENs and vice versa.
   - Secure Token Generation: The final composite token should only be able to be generated by an authenticated vehicle through aggregating the pieces of token shares from multiple ENs. Note that ENs themselves cannot produce a final token even \( t - 1 \) of them collude with each other.
   - Attacks Resistance: It should be secure against the above five attacks.

3) **Scalability**: Scalability means that the proposed protocol should support dynamic property in terms of update of vehicles and ENs.

- **Update of Vehicles**: It should support secure and efficient update of vehicles including joining and revocation. In particular, a newly joined vehicle should be authenticated by ENs as normally as existing ones, while revocation of a vehicle requires that the vehicle cannot request for services anymore even with its previous private keys.

- **Update of ENs**: It should support secure and efficient update of ENs including joining and revocation. Specifically, a newly joined EN should be able to authenticate any vehicle together with other ENs, while a revoked EN cannot be exploited by any adversary to authenticate vehicle users jointly with other ENs.

V. THE PROPOSED THRESHOLD MUTUAL AUTHENTICATION PROTOCOL

A. Design Challenges and Our Ideas

Putting forward a specific protocol that can achieve all the security goals is not trivial. Several design challenges are supposed to be carefully addressed.

- First, the property of multiple ENs verifying a single vehicle provides a natural environment for attackers to launch a flooding attack. Specifically, an attacker could simply flood a large number of requests, each of which will be checked by multiple ENs. This flooding attack will definitely waste the whole network resources. In order to prevent this attack, we propose to set up a leader EN (e.g., the first EN which receives the vehicle’s request) and requires the leader EN to verify the request first before forwarding it to other ENs. If the request is not valid, the leader EN rejects the vehicle immediately and will not forward it. One possible vulnerability of this method is that if the leader EN is compromised then it could be controlled by the attacker to intentionally reject legitimate vehicles’ requests. Fortunately, since vehicles are usually highly moving from one place to another, the duration of service interruption caused by a compromised EN is quite limited. Namely, once the vehicle enters another EN’s communication range, it can request services from the new leader EN.

- In addition, it is also subtle to provide certificateless mutual authentication between vehicles and ENs. In particular, ENs need to authenticate vehicles before providing services, while vehicles also need to ensure that they are requesting services from legitimate ENs. A naive way to achieve this is to employ PKI-based signatures, but the large number of vehicles makes it impractical for each EN to query the certificates of vehicles whenever authenticating them. To address this issue, we adapt identity-based signatures for ENs to authenticate vehicles. On the other hand, for vehicles authenticating ENs, it is acceptable to require vehicles to query for ENs’ certificates since the number of ENs is fractional to that of vehicles.

- Finally, the proposed protocol should also support secure update of vehicles and ENs such as joining or revocation. For example, a revoked vehicle should not be able to utilize its previous secrets to generate a valid token, while
a revoked EN should not be able to have the power to cooperate with other ENs to authenticate any vehicle. We associate a state with each vehicle in order to manage the update of vehicles, while we propose a distributed method to securely update the authentication information stored on ENs in terms of joining or leaving of the ENs, respectively.

B. Protocol Design

We elaborate the proposed authentication protocol which contains setup phase, registration phase, and authentication phase. We divide the authentication phase into two cases: Auth-I which is the first authentication with token generation phase and Auth-II which is the second and future authentication phase due to the requirement of fast handover authentication when a vehicle moves from the radiation coverage of one EN to another. Intuitively, in Auth-I a vehicle V is firstly authenticated by a set of ENs which then collaboratively generate a valid token that can be used as the authentication credential for Auth-II phase. The proposed protocol achieves mutual authentication, i.e., vehicles and ENs can authenticate each other. A vehicle V is authenticated collectively by $T(|T| = t)$ ENs. Since V could be only in the communication range of certain ENs (w.l.o.g., say $F_i$) at any time, it is necessary to require $F_i$ to forward V’s request information to other ENs. $F_i$ thus becomes the leader EN. Specifically, the procedure of Auth-I is elaborated as follows.

1) Setup Phase: In the system setup, a bilinear map $e : G \times G \rightarrow G_T$ is created by RS, where $G$ is an additive cyclic group with order $q$ and $G_T$ is a multiplicative cyclic group with order $q$. Four cryptographic hash functions $H, H_1, H_2$ and $H_3$ are selected, where $H$ is defined as $H : [0, 1]^* \rightarrow G, H_1$ is defined as $H_1 : [0, 1]^* \times G \rightarrow G, H_2$ is defined as $H_2 : G_T \rightarrow G$, and $H_3$ is defined as $H_3 : G_T \rightarrow [0, 1]^*$. Then, RS chooses a random element $x$ from $Z_q^{*}$ as its master secret key and the master public key is $P_{pub} = xP$.

In addition, RS chooses another random element $s$ from $Z_q^{*}$ and divides it into $n$ shares $s_i (1 \leq i \leq n)$ with a $(t, n)$-secret sharing scheme, so that any $t$ or more shares of $s$ can be combined to recover the secret $s$. Note that $s$ is the composite secret key that will be used to compute the composite token and the corresponding public verification key is $PS = sP$. To distribute the secret key $s$, RS selects a random polynomial function with degree of $t - 1$ as Equation 1.

$$f(x) = s + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1} \pmod q$$ (1)

where $a_1, \cdots, a_{t-1} \in Z_q^{*}$ are randomly chosen polynomial coefficients. RS sets $s_i = f(i)$ as the secret key of EN $F_i (i \in [1, n])$ and sends $s_i$ to $F_i$ through a secure channel. The corresponding public key of $F_i$ becomes $PK_{F_i} = s_iP$. RS then publishes these public keys: $P_{pub}, PS$ and $PK_{F_i} (i \in [1, n])$.

2) Registration Phase: In this phase, a vehicle V sends its identity $ID_V$ to RS which computes the corresponding private key for V as $sk_V = xH(ID_V)$. In practice, $ID_V$ could be the VIN (vehicle identification number) that is a unique code used to identify individual vehicles. In addition, for each vehicle, there is a one-bit state $st \in [0, 1]$ which indicates the registration state. Namely, $st = 1$ means that it is registered as a legitimate vehicle while $st = 0$ means it has been revoked. Once the registration is successful, RS will set the vehicle’s state to be 1 and send all pairs of $(ID_V, stV)$ to each of the ENs.

3) Auth-I Phase: This phase includes two steps: a vehicle is firstly authenticated by a set of ENs which then collaboratively generate a valid token that can be used as the authentication credential for Auth-II phase. The proposed protocol achieves mutual authentication, i.e., vehicles and ENs can authenticate each other. A vehicle V is authenticated collectively by $T(|T| = t)$ ENs. Since V could be only in the communication range of certain ENs (w.l.o.g., say $F_i$) at any time, it is necessary to require $F_i$ to forward V’s request information to other ENs. $F_i$ thus becomes the leader EN. Specifically, the procedure of Auth-I is elaborated as follows.

a) It first checks whether the difference between the timestamp $TS$ and the current time is within a given threshold in order to prevent replay attacks. In addition, it also checks the registration state $stV$ of the corresponding vehicle V by searching from its database. If $stV = 1$, then it goes on with the sequential steps.

b) It authenticates V by checking Equation 2. If the checking does not pass, it rejects V immediately.

$$e(Y, P) = e(R_1, P_{pub}) \cdot e(H_1(ID_V||TS||R_1), R_2)$$ (2)

c) It generates a composed string $EXP$ which may include V’s identity/attribute information $ID_V$, the expiration time and a policy that would control the nature of access. Then it computes a signature $\sigma_i$ on $EXP$ by $\sigma_i = s_i \cdot H(EXP)$. Typically, $\sigma_i$ is the partial token generated by $F_i$. Finally, $F_i$ forwards V’s request information req together with $EXP$ to the other ENs.

d) It selects a random element $r_2^i$ from $Z_q^{*}$ and computes $U_i = e(H(ID_V), P_{pub})r_2^i$, $W_i = \sigma_i \oplus H_2(U_i)$, $C = EXP \oplus H_3(U_i)$ and $V_i = r_2^iP$.

e) Meanwhile, for the other ENs $F_j (j \in [1, n], j \neq i)$, we assume a set of at least $t - 1$ ENs are ready to collaboratively authenticate V and generate a valid token together with $F_i$. They do the same actions as $F_i$ to authenticate V and accept the legitimacy of $EXP$. Then they generate and return their individual $V_j$ and $W_j$ to $F_i$. By far, $F_i$ and the other cooperative ENs constitute a set $T(|T| \geq t)$.

f) It sends all the pairs of $(V_k, W_k) (k \in T)$ and $C$ to $V$. It also computes $K = H_3(Y||U_i)$ which is the session key that will be used between it and V for secure communication purpose.
3) Upon receiving the messages from $F_i$, $V$ will do the following actions.

a) It recovers $\text{EXP}$ by $\text{EXP} = C \oplus H_3(e(sk_v, V))$. Note that this holds because $e(sk_v, V) = e(xH(ID_V), r_i^3, P) = e(H(ID_V), P^{\frac{y_2}{x}}) = U_i$.

b) For all $k \in T$, $V$ recovers $\sigma_k = W_k \oplus H_2(e(sk_v, V))$. Then, it checks whether Equation 3 holds which allows $V$ to efficiently authenticate multiple ENs in a batch instead of authenticating them individually. If the batch verification fails, we can employ the “divide-and-conquer” method to identify the invalid signatures. The basic idea is to divide the set of signatures into two subsets and do the batch verification respectively. This procedure is iteratively exceeded until all the invalid signatures are found out. If for some $\ell \in T$, the signature $\sigma_\ell$ is invalid, then it means $F_i$ is illegitimate and thus is rejected. Then $V$ reports this error to $F_i$ so that $F_i$ can ask for another EN for cooperation.

$$e\left(\sum_{k \in T} \sigma_k, P\right) = e(H(\text{EXP}), \sum_{k \in T} PK_{F_k}) \quad (3)$$

c) It computes $\sigma = \sum_{k \in T} \omega_k \sigma_k = \sum_{k \in T} \omega_k H(\text{EXP}) = sH(\text{EXP})$ where $\omega_k = \prod_{\ell \in T, \ell \neq k} \frac{t}{t}$. If $\sigma_k$ is valid, then $\sigma$ must be valid as well since $\sigma$ is computed based on $\sigma_k$ using polynomial interpolation. $\sigma$ is essentially the token that are generated by aggregating $t$ shares of the token from multiple ENs collaboratively. It also computes the session key with $K = H_3(Y|| e(sk_v, V))$. This finishes the session key establishment since $H_3(Y|| e(sk_v, V)) = H_3(Y|| e(xH(ID_V), r_i^3, P)) = H_3(Y|| e(H(ID_V), P^{\frac{y_2}{x}})) = H_3(Y|| U_i)$.

Note that in our scheme, only the vehicle $V$ itself can do the aggregation since each partial token is encrypted with $V$’s public key. This can avoid the attack in which an adversary compromises only one EN and pretends to be a leading EN to collect $t - 1$ pieces of partial tokens from other ENs and consequently obtain the composite token. Actually, even $t - 1$ ENs are compromised, the attacker still cannot generate a valid token.

4) Auth-II Phase: Once a vehicle $V$ finishes the first authentication, it will obtain a token $\sigma$ together with the string $\text{EXP}$ for its future authentication before the expiry date indicated in $\text{EXP}$. Actually, we can consider $\sigma$ as $V$’s secret credential that can be used for future fast authentication purpose while $\text{EXP}$ is the auxiliary information. In the Auth-II phase, $V$ just needs to show that it holds a valid token to its nearby EN upon requesting services. Suppose $V$ that holds a valid token $\sigma$ and $\text{EXP}$ moves into the communication area of a new edge node $F_k$ and requests to connect to the vehicular networks. Then the mutual authentication between $V$ and $F_k$ is elaborated as follows.

1) $V$ selects a random element $r_3$ from $Z_q^*$, and computes $R_3 = r_3 H(\text{EXP}), R_4 = r_3 \sigma + H_1(\text{EXP}||TS||r_3 PK_{F_k}))$, where $TS$ is a timestamp, and “||” means a concatenation operation. Then $V$ sends a request $req = (ID_V, \text{EXP}, Y_2, R_3, R_4, TS)$ to $F_k$. Meanwhile, $V$ also computes the session key as $K = H_3(r_3 PK_{F_k})$.

2) Upon receiving the request information, $F_k$ first checks the validity of $TS$ and the registration state $stv$ of $V$ by searching from its database. If both are valid, it authenticates $V$ by checking Equation 4 with its secret key $s_k$. If the check does not pass, it rejects $V$ immediately. Otherwise, it calculates the session key as $K = H_3(s_k R_4)$.

$$e(Y_2, P) \equiv e(R_3, P_3) \cdot e(H_1(\text{EXP}||TS||s_k R_4), R_4) \quad (4)$$

If everything goes smoothly, $V$ can communicate with $F_k$ securely with the session key $K$, since $K = H_3(r_3 PK_{F_k}) = H_3(r_3 s_k P) = H_3(s_k R_4)$. Note that $V$ can pre-compute the request information just before moving into the new area so that the authentication delay can be further reduced.

C. Correctness Proofs of the Equations

We will prove the correctness of equation 2, equation 3 and equation 4. Namely, ENs can indeed authenticate a vehicle through Equation 2 while each vehicle can aggregately authenticate a bunch of ENs at the same time by Equation 3. On the other hand, in Auth-II phase, $V$ can be fast authenticated by any edge node $F_k$.

First, we have:

$$e(Y, P) = e(r_1 sk_v + H_1(ID_V||TS||R_1), P)$$
$$= e(r_1 xH(ID_V), P) \cdot e(r_1 H_1(ID_V||TS||R_1), P)$$
$$= e(r_1 H_1(ID_V), xP) \cdot e(H_1(ID_V||TS||R_1), r_1 P)$$
$$= e(R_1, P^{\frac{y_2}{x}}) \cdot e(H_1(ID_V||TS||R_1), R_2)$$

which indicates the correctness of Equation 2.

Then, for the correctness of Equation 3, we have

$$e\left(\sum_{k \in T} \sigma_k, P\right) = e\left(\sum_{k \in T} s_k H(\text{EXP}), P\right)$$
$$= e(H(\text{EXP}), \sum_{k \in T} s_k P)$$
$$= e(H(\text{EXP}), \sum_{k \in T} PK_{F_k}).$$

Finally, for the correctness of Equation 4, we have

$$e(Y_2, P) = e(r_3 \sigma + H_1(\text{EXP}||TS||r_3 PK_{F_k}), P)$$
$$= e(r_3 sH(\text{EXP}) + r_3 H_1(\text{EXP}||TS||r_3 PK_{F_k}), P)$$
$$= e(r_3 sH(\text{EXP}), P) \cdot e(r_3 H_1(\text{EXP}||TS||r_3 PK_{F_k}), P)$$
$$= e(r_3 H(\text{EXP}), sP) \cdot e(H_1(\text{EXP}||TS||r_3 s_k P), r_3 P)$$
$$= e(R_3, P_3) \cdot e(H_1(\text{EXP}||TS||s_k R_4), R_4)$$
D. Enabling Update of Vehicles and ENs

The proposed protocol supports secure update of vehicles and ENs where update includes a new entity joining and a current entity revocation. Note that achieving update functionality is not trivial since an improper design could occur security issues. We will elaborate how our protocol can support the update of both vehicles and ENs, and at the same time can avoid potential attacks.

1) Update of Vehicles:
   a) Joining of a Vehicle: Whenever a new vehicle $V$ hopes to join in the system, it can first register with its $ID_V$ to RS and obtain the corresponding secret key $xH(ID_V)$. RS sets its registration state $st_V$ to 1 and sends $(ID_V, st_V)$ to all the $n$ ENs. By far, $V$ joins in the system successfully.

   b) Revocation of a Vehicle: In case that a vehicle $V$ is scrapped or stolen, the owner should notify RS which will set the corresponding registration state $st_V$ to be 0 and sends $(ID_V, st_V)$ to all the ENs. Upon receiving the record, each EN replaces the value of $st_V$ with the new one. By doing this, if there is a request sent from the same $ID_V$, it will be rejected immediately. To avoid infinite growth of the table size on the EN side, we can optionally require RS to associate an expiration time for the revocation list. For example, if the fact that $st_V = 0$ has lasted more than half a year, ENs can remove the related records about $V$ from the database.

2) Update of ENs:
   a) Joining of an EN: When a new EN requests to join in the authentication server group, it needs to obtain its secret share of $s$. The total number of ENs becomes $n + 1$ and the threshold value becomes $t'$, which results in a $(t', n+1)$-secret sharing. We will show how the secret $s$ can be redistributed from original $n$ shares to new $n + 1$ shares from $(t, n)$-sharing to $(t', n+1)$-sharing) without the involvement of RS.

   - First, all of the $n + 1$ ENs interact with each other to negotiate a common set $T(|T| = t)$ from original $n$ ENs.
   - Each EN $F_i$ in $T$ divides its secret share $s_i$ into $n + 1$ pieces $s_{ij}(j \in [1, n + 1])$ with the following polynomial function
     
     $$f(x) = s_i + b_{i1}x + b_{i2}x^2 + \cdots + b_{i(t'-1)}x^{t'-1} \mod q$$

     where $b_{i1}, \cdots, b_{i(t'-1)} \in Z_q^*$ are randomly chosen coefficients. Then $F_i$ sends $s_{ij} = f(j)$ to EN $F_j(j \in [1, n + 1], j \neq i)$.

   - When $F_j$ receives $s_{ij}$ from all the ENs in the set of $T$, it can reconstruct a new secret share $s_j'$ of $s$ by Equation 6.

     $$s_j' = \sum_{i \in T} \prod_{\ell \in T, \ell \neq i} s_{ij}$$

   By far, all the ENs in the new authentication group constitute a $(t', n+1)$-sharing structure, where any $t'$ out of $n + 1$ ENs can collaborate to authenticate a vehicle and generate a valid token.

   b) Revocation of an EN: When an EN is revoked with some reason, the secret keys should also be redistributed among the rest ENs so that the revoked EN cannot take part in authenticating vehicles anymore. In this case, the total number of ENs becomes $n - 1$ and the threshold value becomes $t''$, which results in a $(t'', n-1)$-secret sharing. We will show how the secret $s$ can be redistributed from original $n$ shares to new $n - 1$ shares from $(t, n)$-sharing to $(t'', n-1)$-sharing).

   - First, all of the $n - 1$ ENs interact with each other to negotiate a common set $T(|T| = t)$ of ENs.
   - Each EN $F_i$ in $T$ divides its secret share $s_i$ into $n - 1$ pieces $s_{ij}(j \in [1, n - 1])$ with the following polynomial function
     
     $$f(x) = s_i + c_{i1}x + c_{i2}x^2 + \cdots + c_{i(n-1)}x^{n-1} \mod q$$

     where $c_{i1}, \cdots, c_{i(n-1)} \in Z_q^*$ are randomly chosen coefficients. Then $F_i$ sends $s_{ij} = f(j)$ to EN $F_j(j \in [1, n - 1], j \neq i)$.

   - When $F_j$ receives $s_{ij}$ from all the ENs in the set of $T$, it can reconstruct a new secret share $s_j''$ of $s$ by Equation 8.

     $$s_j'' = \sum_{i \in T} \prod_{\ell \in T, \ell \neq i} s_{ij}$$

By far, all the ENs in the new authentication group constitute a $(t'', n-1)$-sharing structure, where any $t''$ out of $n - 1$ ENs can collaborate to authenticate a vehicle and generate a valid token.

E. Discussion

In the proposed protocol, we assume that the registration server RS is a trusted party for simplicity. However, in practice, RS could represent a single point of failure. We discuss why we make this assumption and explain how to remove it. On one hand, RS is in charge of registration and revocation for vehicles and edge nodes, and thus it can be offline during the authentication phase which happens most frequently. Less time when connected with the networks provides attackers less transaction records which makes it more difficult for attackers to compromise RS. On the other hand, the single-point-of-failure issue could also be mitigated with a threshold cryptosystem. Namely, RS could be split into multiple sub-registration servers (Sub-RSs) which collaboratively take...
charge of the registration and revocation jobs. The master secret key can be distributed to these Sub-RSs with secret sharing schemes without a trusted party, where each Sub-RS can select its own secret key share and a threshold number of shares can be aggregated into a composite secret that serves as the master secret key. The key distribution of a secret sharing scheme without a trusted dealer has been well studied for many years and is not the highlight of this article, so we neglect the details and assume that RS is a trusted party.

VI. SECURITY ANALYSIS

In this section, we analyze the security of the proposed threshold mutual authentication protocol. In particular, we will discuss how the proposed protocol under the edge-assisted decentralized architecture can achieve all the security goals listed in section IV-C.

A. Mutual Authentication

The proposed protocol can achieve mutual authentication, i.e., vehicles and ENs can authenticate each other. We will discuss the details separately.

1) Authentication of Vehicles: In Auth-I phase, to authenticate a vehicle \( V \), an EN first computes \( H_1(ID_V || TS || R_1) \) upon receiving \( req \) and then checks whether Equation 2 holds. Only a legitimate vehicle with a valid secret key \( sk_V = xH(ID_V) \) can generate a valid verifiable proof \( Y \). An adversary without \( sk_V \) has to forge a valid \( Y^* = x^*H(ID_V) + r^*H_1(ID_V || TS || R_1^2) \) as well as \( R_1^2 = r^*P \) so that the proof can satisfy Equation 2. Even with the ability to eavesdrop previous transactions such as \( req \) that includes \((Y, R_1, R_2)\), it cannot recover \( sk_V \) since \( r_1 \) is impossible to be recovered due to the intractability of the DL assumption. This essentially requires the adversary to construct a valid \( x^*H(ID_V) \) without knowing \( xH(ID_V) \), the probability of which is obvious negligible. Thus, the authentication of vehicles can be achieved. Similarly, in Auth-II phase, the adversary has to construct a valid \( s_{r_2} H(EXP) \) without knowing \( sH(EXP) \) in order to pass Equation 4, which again violates the DL assumption. Thus, the authentication of vehicles can be achieved.

2) Authentication of ENs: In Auth-I phase, to authenticate an EN \( F_i \), a vehicle checks whether Equation 3 holds. Specifically, \( \sigma_i \) is a signature generated by \( F_i \). Only a legitimate EN with the corresponding private key \( s_i \) can produce a valid signature. Since the signature algorithm is based on the BLS short signature [49], the security analysis of it can also be impossible to be recovered due to the intractability of the DL assumption. This essentially requires the adversary to construct a valid \( x^rH(ID_V) \) without knowing \( xH(ID_V) \), the probability of which is obvious negligible. Thus, the authentication of vehicles can be achieved. Similarly, in Auth-II phase, the adversary has to construct a valid \( s_{r_2} H(EXP) \) without knowing \( sH(EXP) \) in order to pass Equation 4, which again violates the DL assumption. Thus, the authentication of vehicles can be achieved.

V. SECURITY ANALYSIS

In this section, we analyze the security of the proposed threshold mutual authentication protocol. In particular, we will discuss how the proposed protocol under the edge-assisted decentralized architecture can achieve all the security goals listed in section IV-C.

A. Mutual Authentication

The proposed protocol can achieve mutual authentication, i.e., vehicles and ENs can authenticate each other. We will discuss the details separately.

1) Authentication of Vehicles: In Auth-I phase, to authenticate a vehicle \( V \), an EN first computes \( H_1(ID_V || TS || R_1) \) upon receiving \( req \) and then checks whether Equation 2 holds. Only a legitimate vehicle with a valid secret key \( sk_V = xH(ID_V) \) can generate a valid verifiable proof \( Y \). An adversary without \( sk_V \) has to forge a valid \( Y^* = x^rH(ID_V) + r^*H_1(ID_V || TS || R_1^2) \) as well as \( R_1^2 = r^*P \) so that the proof can satisfy Equation 2. Even with the ability to eavesdrop previous transactions such as \( req \) that includes \((Y, R_1, R_2)\), it cannot recover \( sk_V \) since \( r_1 \) is impossible to be recovered due to the intractability of the DL assumption. This essentially requires the adversary to construct a valid \( x^rH(ID_V) \) without knowing \( xH(ID_V) \), the probability of which is obvious negligible. Thus, the authentication of vehicles can be achieved. Similarly, in Auth-II phase, the adversary has to construct a valid \( s_{r_2} H(EXP) \) without knowing \( sH(EXP) \) in order to pass Equation 4, which again violates the DL assumption. Thus, the authentication of vehicles can be achieved.

2) Authentication of ENs: In Auth-I phase, to authenticate an EN \( F_i \), a vehicle checks whether Equation 3 holds. Specifically, \( \sigma_i \) is a signature generated by \( F_i \). Only a legitimate EN with the corresponding private key \( s_i \) can produce a valid signature. Since the signature algorithm is based on the BLS short signature [49], the security analysis of it can also be impossible to be recovered due to the intractability of the DL assumption. This essentially requires the adversary to construct a valid \( x^rH(ID_V) \) without knowing \( xH(ID_V) \), the probability of which is obvious negligible. Thus, the authentication of vehicles can be achieved. Similarly, in Auth-II phase, the adversary has to construct a valid \( s_{r_2} H(EXP) \) without knowing \( sH(EXP) \) in order to pass Equation 4, which again violates the DL assumption. Thus, the authentication of vehicles can be achieved.

B. Secure Token Generation

In the proposed protocol, a final composite token \( \sigma \) is generated by combing at least \( t \) pieces of token shares \( \sigma_i \) from the corresponding EN \( F_i \) \((i \in T)\) with threshold signature technique, i.e., \( \sigma = \sum_{i \in T} \omega_i \sigma_i \). Each of the piece of token \( \sigma_i \) is encrypted upon sent to the vehicle \( V \), i.e., \( W_i = \sigma_i \oplus H_2(U_i) \) where \( U_i = e(H(ID_V), P_{pub})^2 \). Since \( \sigma_i \) is encrypted with the public key of \( V \), only \( V \) with the valid private key \( sk_V \) can decrypt the message by \( \sigma_i = W_i \oplus H_2(e(sk_V, V_i)) \). If the BDH assumption holds, the encryption algorithm is secure. Thus, only the authenticated \( V \) can obtain all the \( t \) pieces of token shares which are further aggregated into the final composite token. Note that even up to \( t-1 \) ENs collude with each other, they cannot generate the composite token if the \((t, n)\) threshold signature is secure which will be proven in Section VI-C.3.

C. Attacks Resistance

The proposed protocol can mitigate all the five attacks defined in Section IV-B. The detailed analysis is given as follows.

1) Replay Attack: To launch a replay attack, an attacker eavesdrops previous transcripts and tries to pass a new authentication session with the recorded transcripts. In particular, it records \((ID_V, Y, R_1, R_2, TS)\) and sends the tuple to an EN \( F_i \). Note that the proof \((Y, R_1, R_2)\) cannot pass the authentication since \( Y \) includes the timestamp \( TS \) which has already expired. Thus, the proposed protocol is resistant to replay attack.

2) Impersonation Attack: To impersonate a legitimate vehicle \( V \), an attacker has to generate a valid tuple of proof \((ID_V, Y, R_1, R_2, TS)\). Without the secret key \( sk_V \), it cannot produce a valid \( Y \) though. On the other hand, it could also forge a \( Y' \) based on previous \( Y \) such as \( Y' = r'Y = r'Y_1(sk_V + H_1(ID_V || TS || R_1)) \). However, in this case, \( Y' \) includes an expired timestamp \( TS \) that can be detected by the ENs. Therefore, the attacker has no way to forge a valid proof. Namely, the impersonation attack can be effectively prevented.

3) EN Compromise Attack: If the \((t, n)\) threshold signature is secure, then the proposed protocol can tolerate compromise of up to \( t-1 \) ENs. Namely, the attacker has secret keys of these ENs and tries to collaboratively authenticate a vehicle \( V \) and generate a valid access token \( \sigma \) for \( V \). We will prove that even with this secret keys, the attacker cannot forge a valid token.

Without loss of generality, we assume that the \( t-1 \) compromised ENs are \( (F_1, \ldots, F_{t-1}) \) with secret keys \( (s_1, \ldots, s_{t-1}) \). The attacker can thus produce \( t-1 \) pieces of tokens \( \sigma_i = s_i \cdot H(EXP)(1 \leq i \leq t-1) \). If the attacker can produce a valid token \( \sigma = \sum_{i=1}^{t-1} \omega_i \sigma_i \), then it can compute \( \sigma_i = (\omega_i)^{-1} \cdot (\sigma - \sum_{i=1}^{t-1} \omega_i \sigma_i) = (\sum_{j=1}^{t-1} \prod_{j \neq i} i \cdot \prod_{j \neq i} j). This means that the attacker can compute the valid \( \sigma_i \) which should actually be \( s_i \cdot H(EXP) \) without \( s_i \). Namely, given \( H(EXP) \in G, P, s_i P \), the attacker can compute \( s_i \cdot H(EXP) \), which contradicts with the CDH assumption. Therefore, the attacker cannot produce a valid token with up to \( t-1 \) compromised ENs, i.e., our protocol is still secure even up to \( t-1 \) ENs are compromised.
TABLE I
COMPARISON BETWEEN THE PROPOSED PROTOCOL AND EXISTING PROTOCOLS

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Computational Cost</th>
<th>Authentication Delay</th>
<th>SPF Resistance*</th>
<th>F/ENCA Resistance**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azees et al. [17]</td>
<td>$2mt_p + (6t - 2)mt_{sm}$</td>
<td>$2mt_p + 5mt_{sm}$</td>
<td>$m(t_V + t_R + T_{VR})$</td>
<td>$\times$</td>
</tr>
<tr>
<td>Cui et al. [33]</td>
<td>$4mt_{sm}$</td>
<td>$6mt_{sm}$</td>
<td>$m(t_V + t_R + T_{VR})$</td>
<td>$\times$</td>
</tr>
<tr>
<td>Ma et al. [13]</td>
<td>$3mt_{sm}$</td>
<td>$12mt_{sm}$</td>
<td>$m(t_V + t_R + T_{RS})$</td>
<td>$\times$</td>
</tr>
<tr>
<td>Our protocol</td>
<td>$(t + 2)t_p + (4m - 1)t_{sm}$</td>
<td>$(3m + 1)t_p + (m + 2)t_{sm}$ + $(m + 3)t_h$</td>
<td>$t_V + t_R + m(t_R + T_{VR})$</td>
<td>$\checkmark$</td>
</tr>
</tbody>
</table>

* SPF: single point of failure
** F/ENCA: Fog/Edge Node Compromise Attack
\(n\): the number of authentication execution
\(\ell, t\): system parameters (constant)

4) Wasting Network Resources Attack: To launch a wasting network resources attack, an attacker keeps sending a large number of illegal requests to a specific EN which will forward the requests to other ENs for collaborative authentication. This will definitely waste the network resources. In the proposed protocol, whenever an EN receives a request from a vehicle, it first authenticates the request information before forwarding it to other ENs. If the authentication fails, it will reject the vehicle immediately without forwarding. This can effectively mitigate the wasting network resources attack.

5) Token Forgery Attack: To launch a token forgery attack, an attacker tries to forge a token. In the proposed protocol, a token is generated based on threshold signature technique. To generate a valid token, the attacker needs to obtain the secret keys of at least \(t\) ENs. However, due to the hardness of the DL problem, it is computationally infeasible to reveal the secret key \(s_i\) from \(PK_{F_i} = s_i P\). Thus, the proposed protocol can prevent token forgery attack.

D. Secure Update of Vehicles

Secure update of vehicles can be achieved by associating each vehicle with a one-bit state \(st_V\). When a new vehicle \(V\) joins the system, upon registration, \(st_V = 1\) is sent together with the identifier \(ID_V\) to all the \(n\) ENs. When a vehicle \(V\) is revoked (e.g., because of being scrapped or stolen), the owner notifies RS which will set \(st_V = 0\) and send the updated state to all the ENs. Note that the communication between RS and ENs is based on a secure channel, so no attacker can obtain or modify the transcripts.

E. Secure Update of ENs

Secure update of ENs can be achieved by a distributed threshold key management which does not require the involvement of RS. First, we prove that in the case of a new EN joining the system, the constructed \((t', n + 1)-\)sharing is still a sharing of the secret key \(s\) as follows.

According to the construction, \(s = \sum_{i \in T} o_i s_i\) and \(s'_j = \sum_{i \in T} o_i s_{ij}\) where \(o_i = \prod_{\ell \in T, \ell \neq i} \frac{\ell}{\ell - j}\). Since each \(s_i\) is divided into \(n + 1\) pieces by \(F_i\), we can obtain that \(s_i = \sum_{j \in T'} o_j s_{ij}\) where \(o_j = \prod_{\ell \in T', \ell \neq j} \frac{\ell}{\ell - j}\). Therefore, we have

\[\sum_{j \in T'} o_j s_{ij}' = \sum_{j \in T'} o_j \left(\sum_{i \in T} o_i s_{ij}\right) = \sum_{i \in T} o_i \left(\sum_{j \in T'} o_j s_{ij}\right) = \sum_{i \in T} o_i s_i = s\]

As a consequence, any \(t'\) out of \(n + 1\) ENs can reconstruct \(s\). On the other hand, any set of less than \(t'\) ENs cannot reconstruct \(s\) since \(s_i\) is divided into \(\left(t', n + 1\right)\)-sharing and any set of less than \(t'\) partial keys \(s_{ij}\) cannot reconstruct \(s_i\). Thus, \((t', n + 1)-\)sharing is still a sharing of the secret key \(s\) for any subset of \(t'\) out of \(n + 1\) ENs including the newly joined one can authenticate vehicles collaboratively.

For the case of an EN \(F_i\) being revoked, the constructed \((t'', n - 1)\)-sharing can be proved to be a sharing of the secret key \(s\) similarly. Since each of the secret shares of \(s\) has been replaced with new share, \(F_i\)'s old secret share \(s_i\) will be invalid and thus \(F_i\) cannot be exploited by attackers to authenticate vehicles jointly with other ENs. That is, the proposed protocol supports secure revocation of ENs as well.

VII. PERFORMANCE EVALUATION

In this section, we analyze and evaluate the performance of the proposed protocol in terms of computation cost and latency induced by the cooperation of ENs. We also compare the performance of the proposed protocol with existing related works. As shown in Table I, we compare the performance of our proposed protocol with existing protocols from several aspects: the computational cost, the authentication delay, and the security functionalities. Let \(t_p, t_{sm}, t_h\) represent the time required to perform a pairing, a scalar multiplication, and a map-to-point hash, respectively, which dominate the computational costs in these protocols. Let \(t_V, t_R, T_{RS}\) be the delay introduced by computation time on vehicle, edge/fog, and AS side respectively, and \(T_{VR}, T_{RS}\) be the communication latency caused by message propagating between the vehicle and edge/fog node, and between the edge/fog node and AS, respectively. In the proposed protocol, the authentication cost in two scenarios is different, i.e., the cost in Auth-I phase is bigger than that if Auth-II phase. However, before the token
Fig. 3. Computation cost in Auth-I phase.

EXP expires, the Auth-I phase is only performed once. Thus, with the number of authentication execution \(m\) increasing, the average cost decreases. Table I shows the computational cost on both vehicle and fog/edge/AS side for a number of \(m\) authentication execution. It indicates that the proposed protocol does not have to be the most computationally efficient one. Nevertheless, our protocol stands out in the authentication delay and security aspects. Namely, with \(m\) increasing, the authentication delay increases much more slowly than the other protocols. In addition, our protocol can prevent both single point of failure and edge node compromise attack. We will give detailed analysis and performance evaluation subsequently.

The experiment is conducted with a laptop owning a 2.8GHz Intel(R) Core(TM) i7-7700HQ CPU and 16 GB of RAM. We use Java language to write the programming code which invokes the Java Pairing-Based Cryptography Library (JPBC) with version of 2.0.0. The Type-A pairing built on the curve \(y^2 = x^3 + x\) with the embedding degree of 2 is chosen. The corresponding group \(G\) has order of \(q\) with \(\log q = 160\) and the size of a group element is 512 bits. In order to increase the accuracy, we run each of the evaluation 50 times and take the averaged values as the final results.

A. Computation Cost

We evaluate the time cost required to do the authentication on both vehicles and ENs’ sides. For the \((t, n)\)-threshold signature, we evaluate the scenarios of \(t\) up to 10. This is reasonable, since it is really hard for an attacker to compromise more than 10 ENs within a short period (in practice we can require the ENs to update their private keys daily or weekly.)

1) Cost in Auth-I Phase: First, we evaluate the performance of Auth-I phase, where a vehicle needs to authenticates a set of ENs and aggregate at least \(t\) partial tokens into the composite token. The vehicle needs to perform \((t+2)\) bilinear pairing operations, 3 scalar multiplications, and \((t+3)\) map-to-point hashes, while an edge node needs to perform 4 bilinear pairing operations, 3 scalar multiplications, and 4 map-to-point hashes, respectively. To evaluate the time to generate a token on the vehicle side, we conduct experiments with different number of cooperative ENs that authenticate the vehicle, that is, we consider a \((t, n)\)-secret sharing scheme with \(t \in [1, 10]\) and a fixed \(n = 10\). The result is shown in Figure 3(a). Even in the case of \(t = 10\), it only costs 33ms to generate a valid token. In addition, we also explore the impact of the \(t\) and \(n\) on the computation cost in Auth-I phase. As shown in Figure 3(b), with the same threshold \(t\), distinct values of \(n\) result in similar runtime, which is aligned with our design. Namely, for a given \(t\), the vehicle only needs to authenticate \(t\) ENs and combine \(t\) pieces of partial tokens, and thus the authentication time should be roughly the same. With a higher \(t\), the computation cost increases as well which is also reasonable. Finally, we evaluate the authentication cost of both vehicles and ENs. As shown in Figure 3(c), the authentication time for vehicles is different when the number of cooperative ENs (i.e., \(t\)) required to authenticate vehicles changes, since vehicles need to authenticate all \(t\) ENs. However, since our scheme supports batch verification for the vehicle to authenticate multiple ENs, the number of communication pairing operations does not increase with the growth of \(t\). Even when \(t = 10\), the authentication time per vehicle is still less than one second (641 ms). On the other hand, the authentication cost of each EN keeps constant because it only needs to verify one vehicle per request.

2) Cost in Auth-II Phase: As mentioned before, in the proposed protocol, once the vehicle obtains a token, it can hold and utilize this token for its future authentication purpose before the token expires. Thus, for the second-and-later authentication, only the vehicle and its nearest EN run the protocol, which significantly alleviates the computational burden of both vehicles and ENs. The vehicle needs to perform 4 scalar multiplications, and 1 map-to-point hash, while an edge node needs to perform 3 bilinear pairing operations, 1 scalar multiplications, and 1 map-to-point hash, respectively.

3) Overall Computation Cost: According to the above analysis, the overall computation cost can be simply derived. Moreover, since our protocol support fast handover authentication, we will evaluate the overall computation cost over multiple authentication executions. Let \(m\) be the number of authentication executions, then the overall computation cost of our protocol on vehicle and EN side is \((t+2)tp+(4m-1)ttn+(m+t+2)tth\) and \((3m+1)tp+(m+2)ttn+(m+3)tth\), respectively. Without loss of generality, we consider \(t = 5\), i.e., at
Fig. 4. Comparison of Overall Computation Cost. Without loss of generality, we consider $t = 5$, i.e., at least 5 ENs cooperatively authenticate a vehicle. We set the parameter $\ell = 5$ in [17].

Fig. 5. Performance evaluation and comparison of the authentication latency.

At least 5 ENs cooperatively authenticate a vehicle. We evaluate the overall computation cost including the vehicle and the EN authentication of our protocol and existing protocols, respectively. Figure 4 shows the result which demonstrates that the computational efficiency of our protocol approximates to that of the most efficient one [33].

B. Authentication Latency

Authentication latency means the delay caused by the mutual authentication, which contains the computation time (as mentioned before, may include $t_V, t_R, t_S$) and message propagation time (may include $T_{VR}, T_{RS}$). Due to the distinguished operations in different authentication phases, the latency also varies. In Auth-I phase, since authenticating a vehicle requires the cooperation of at least $t$ ENs at different locations, it may induce latency including the computation time on the ENs side (denoted as $t_R$) and the vehicle side (denoted as $t_V$). For simplicity, we assume all the cooperative ENs can simultaneously deal with requests that are transmitted by the leader EN. Then the latency becomes $t_V + 2t_R$, which contains $t_R$ caused by the leader EN and $t_R$ caused by the cooperative ENs.

In Auth-II phase, the vehicle can pre-compute the credentials based on its token before it moves to the communication area of a new EN and thus the authentication latency caused on the vehicle side $t'_V$ could be ignored (i.e., $t'_V = 0$). Likewise, ENs do not need to generate partial tokens either in this phase and thus the latency $t'_R$ is also reduced. Thus, for a number of $m$ authentication executions, the overall authentication delay of our protocol becomes $t_V + t_R + m(t'_R + T_{VR})$. The overall authentication delay of existing protocols can be found in Table I. Although Ma et al.’s protocol [13] requires least computation resources on vehicles, its authentication architecture belongs to Type 1. That is, in their protocol, vehicles are authenticated by a remote authentication server, which increases the authentication delay.

To evaluate the authentication latency, we assume that $T_{VR} = 100 \text{ ms}$ and $T_{RS} = 40 \text{ ms}$, which represents the communication latency of a typical V2I communication [50].
and a typical service provider, respectively. We consider the parameters $t = \ell = 5$. We first evaluate the performance of our protocol, as shown in Figure 5(a). Except for the overall authentication latency, we also compute the average authentication latency. It can be found that the average authentication latency is gradually reduced with $m$. Moreover, we compare the authentication latency of our protocol with existing protocols as shown in Figure 5(b). The result indicates that when $m$ becomes larger than 9, the overall authentication latency of our protocol is the smallest, mainly because of the fast handover authentication of our protocol. In practice, the number of $m$ is easily larger than 100. Considering the coverage range of an RSU is within 500 meters, suppose the velocity of a vehicle is 80Km/h, then the time that the vehicle connects to the RSU is less than 20 seconds. Thus, a vehicle will roam to new RSUs for about 180 times per hour if it hopes to enjoy seamless network services. Note that the evaluated authentication latency varies on different devices with distinct running environment. Specifically, in our simulated experiment, we use Java language to implement the code. The execution time of one bilinear pairing operation $t_{p}$, one scalar multiplication $t_{sm}$, and one map-to-point hash $t_{h}$ is about 24 ms, 15 ms and 25 ms, respectively. In other papers which use similar personal computer to implement the code with C language [33], the corresponding execution time of these operations is 4.211 ms, 1.709 ms, and 4.406 ms, respectively. Therefore, if implementing with their code, the authentication latency will be further reduced. For instance, with $t = 5$, the authentication latency for Auth-I scenario is about 200 ms, while the authentication latency for Auth-II scenario is about 130 ms. As a result, our protocol caters for highly mobile scenarios like vehicular networks.

VIII. CONCLUSION

In this article, we have investigated the access authentication issues for vehicular networks and proposed an edge-assisted decentralized mutual authentication protocol, in which vehicles and edge nodes can efficiently authenticate each other with only one-round interaction. In the first authentication, each vehicle can be collaboratively authenticated by multiple edge nodes which then generate an access token all together based on $(t, n)$ threshold signature for the vehicle that can be used for fast handover authentication later on. We have evaluated the performance of the protocol and the results show that our protocol can reduce the authentication delay to a big extent, and the fast handover authentication further reduces the authentication latency significantly. For instance, after executing the authentication for ten times, the overall authentication latency is less than 2 seconds and the average latency is reduced to less than 0.2 second. These advantages should expedite the deployment of intelligent transporta-

REFERENCES


[2] S. Chen et al., “Vehicle-to-everything (V2X) services supported by LTE-

[3] X. Shen et al., “AI-assisted network-slicing based next-generation wire-


[10] S. Chen et al., “Vehicle-to-everything (V2X) services supported by LTE-


Cheng Huang (Member, IEEE) received the B.Eng. and M.Eng. degrees from Xidian University, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada. He was a Project Officer of the INFINITUS Laboratory, School of Electrical and Electronic Engineering, Nanyang Technological University, till July 2016. His research interests include applied cryptography, cyber security, and privacy in the mobile networks.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada. His research interests include network resource management, wireless network security, the Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks. He is a registered Professional Engineer of Ontario, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and the Communications Society. He received the R. A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals, ON, in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and the Education Award in 2017 from the IEEE Communications Society, the Technical Recognition Award from the Wireless Communications Technical Committee in 2019 and the AHSN Technical Committee in 2013, the Excellent Graduate Supervision Award in 2006 from the University of Waterloo, and the Premier’s Research Excellence Award (PREA) in 2003 from the Province of Ontario. He has served as the Technical Program Committee Chair/Co-Chair of the IEEE GLOBECOM’16, the IEEE Infocom’14, the IEEE VTC’10 Fall, and the IEEE GLOBECOM’07 and the Chair of the IEEE Communications Society Technical Committee on Wireless Communications. He is the elected IEEE Communications Society Vice President for Technical and Educational Activities, the Vice President for Publications, a Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of the IEEE ComSoc Fellow Selection Committee. He was/is the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, IEEE Network, IET Communications, and Peer-to-Peer Networking and Applications.