IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 October 2022, accepted 2 November 2022, date of publication 14 November 2022, date of current version 22 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3222302

RESEARCH ARTICLE

Dual Privacy-Preserving Lightweight Navigation System for Vehicular Networks

YINGYING YAO[1,2], XIAOLIN CHANG[1,2], (Senior Member, IEEE), LIN LI[1], JIQIANG LIU[1], (Senior Member, IEEE), AND HONG WANG[2]
[1]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China
[2]Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, Henan 450001, China

Corresponding author: Xiaolin Chang (xlchang@bjtu.edu.cn)

This work was supported by the Henan Key Laboratory of Network Cryptography Technology under Grant LNCT2020-A07.

ABSTRACT As an indispensable part of intelligent transportation system, a traffic-sensitive navigation system can assist drivers in avoiding traffic congestion by providing navigation services. The navigation service provider (NSP) utilizes road condition information from nearby vehicles collected by roadside units (RSUs) to guide a vehicle through an optimal route provided by the navigation services. This paper mainly focuses on tackling three key issues in such navigation system. Firstly, it is essential to ensure the privacy of the vehicles' personal identifiable information (PII). Secondly, the location, start point destination and route of the vehicle should not be leaked to RSUs and also should not be linked to its PII by RSUs and NSP. Thirdly, the process of vehicles obtaining the navigation services should be lightweight. Because of such problems, this paper proposes a dual privacy-preserving lightweight navigation system for vehicular networks through designing a novel signature scheme and combining other cryptographic primitives to keep dual privacy including PII and route related information. And the correctness analysis of the proposed system, security proof of the designed signature scheme adopted in the proposed system, and privacy analysis of the system are thoroughly provided. In addition, the performance of the proposed system is evaluated and compared with the existing systems to illustrate that the proposed system is efficient.

INDEX TERMS Vehicular networks, navigation system, dual privacy-preserving, security, lightweight.

I. INTRODUCTION

According to the latest prediction of the United Nations, by 2050, more than half of the world's population will live in urban areas [1]. Vehicle pollution emission, transportation delay and traffic congestion are pervasive phenomena in highly urbanized cities [2], which will significantly affect people's lives. In order to prevent and alleviate these issues, the concept of smart city is proposed, which is designed, constructed, and maintained by using highly advanced integrated technology, and it will provide an intelligent platform to integrate technology into society [3]. Intelligent transportation system as an important part of smart city, vehicular networks provide popular vehicular applications like parking navigation [4], parking spot finding [5], road monitoring [6], and ride-hailing [7], [8], [9] to alleviate traffic congestion and missions, support transportation infrastructure costs and manage requirements. These applications usually require the location of vehicles located by global navigation satellite system (GNSS) and traffic related information collected by roadside units (RSUs) to provide location-based services.

To avoid ongoing traffic congestion, traffic-sensitive navigation systems [10] in vehicular networks emerge to provide real-time predictive optimal navigation routes by utilizing the real-time traffic related information. To access the navigation services provided by traffic-sensitive navigation systems, the vehicle needs to submit a requesting query including its current location as enabled by GNSS positioning technology, its traveling speed and destination to its nearby RSU [11]. The RSU will forward it to the navigation service provider (NSP). Then NSP will reply to the optimal route of the vehicle in combination with the real-time traffic related information collected by RSUs. For example, the popular navigation

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

application Google Maps [12] has attracted more than one billion active users around the world.

However, vehicle users are increasingly worrying about their privacy in these traffic-sensitive navigation systems [11], [13]. Many schemes like [14], [15], [16], [17], [18], [19], [20], [21], and [22] have been put forward to protect the privacy of vehicle users. But there still remain some issues in these schemes. They either mainly focused on anonymous authentication to preserve the identity privacy of vehicles, or mainly focus on location privacy of vehicle users, but ignored their navigation destination or route privacy. Although there exist schemes considering the identity, location and route privacy, the destination is leaked to one or more curious-but-honest RSUs and the route is not updated in real time according to current traffic related information. Therefore, the following **challenges** concerning preserving the privacy of vehicles in navigation systems should be considered.

- **Identity privacy.** While a querying vehicle submits the navigation query to its nearby RSU for optimal navigation query, the vehicle's identity related information should not be leaked to the RSU and NSP. That is, the vehicle is anonymous during the navigation query. At the same time, the navigation queries at different RSUs should not be analyzed and linked to the same vehicle if the RSUs collude with each other.
- **Location privacy.** The current location of a vehicle should not be leaked to RSUs and NSP during the vehicle's navigation query. Or the RSUs and NSP cannot link the location information to a specific vehicle.
- **Start point and destination privacy.** The start point and destination of a vehicle's navigation query should not be leaked to RSUs and NSP during the vehicle's navigation query. Or the RSUs and NSP cannot link the start point, and destination of a query to a specific vehicle.
- **Route privacy.** The route information in the navigation service should not be leaked to RSUs. Moreover, NSP cannot link the route privacy to a specific vehicle.
- **Unlinkability.** Expect that RSUs and NSP cannot link the location, start point, destination and route during a navigation query to a specific vehicle as mentioned above. For the same destination queries, NSP cannot distinguish whether they come from the same querying vehicle.
- **Authentication.** During the navigation service provision, the RSUs should verify the validity and legitimacy of a vehicle's identity.
- **Lightweight.** The overhead of the navigation service provision is preferably lightweight. At the same time, NSP should update the congestion situation of the route for the querying vehicle in real time according to the real-time traffic related information received from RSUs.

Aiming to solve the above challenges, this paper proposes a dual privacy-preserving lightweight navigation system for vehicular networks. It realizes the dual privacy preserving of PII, the current location, start point, destination and route information not only for outside attackers but also for the curious-but-honest RSUs and NSP. In addition, the system realizes the computational efficiency for vehicles. Therefore, the proposed navigation system is friendly for the vehicle users with privacy concerns.

Contributions. The key contributions of this paper can be summarized as follows.

- We propose a dual privacy-preserving lightweight navigation system for vehicular networks through designing a novel signature scheme (**Algorithm 1** and **Algorithm 2** in Section V). The system realizes the unlinkable anonymity of vehicles for outsiders and inside RSUs and NSP to preserve the PII privacy, and also realizes unlinkability between the location, start point, destination, route information and PII to preserve the privacy of the location, start point, destination and route information.
- The proposed system adopts uncomplicated cryptographic primitives to realize lightweight navigation service provision for vehicles.
- We provide the correctness, security and privacy analysis to illustrate that the proposed system is correct, and it satisfies the security and privacy requirements.
- We evaluate and compare the overhead of our proposed navigation system with the existing systems to demonstrate that the computing overhead of our proposed system is efficient.

Organization. The rest of the paper is organized as follows. The related work is presented in Section II. The preliminaries are introduced in Section III. Section IV provides the overview of system model, threat model and security model. The pro-posed navigation system is detailed in Section V. The correctness, security and privacy analysis are given in Section VI. Section VII explains the performance evaluation. The final section summarizes the paper and introduces our future work.

II. RELATED WORK

In recent years, a large body of literature on preserving the privacy of navigation systems for vehicular networks has emerged.

Ni et al. [4] proposed P-SPAN for vehicular ad hoc network, which is a privacy-preserving smart parking navigation system through combining vehicular communications and cloud storage to provide secure smart parking navigation services for vehicle users. In P-SPAN scheme, a vehicle user needs to submit its current location and destination to the cloud service provider to get the real-time parking information. Thus, the cloud service provider can obtain the identity, location and destination privacy of the vehicle.

Zhu et al. [5] proposed ASAP, which is an anonymous smart-parking and payment scheme in vehicular networks. ASAP provides the private parking spots to save the fuel and time of vehicles and alleviate the traffic congestion. In the scheme, the vehicles' real locations were hidden and mapped to a wide area, but the real location and route can be revealed when the similar navigation queries are linked.

TABLE 1. **Comparison of privacy properties.**

Property		[4] 2018	[5] 2018	[23] 2018	[24] 2019	[25] 2020	[26] 2021	[27] 2022	Our proposed system
Identity Privacy	from outsiders	√	√	n/a	×	√	√	n/a	√
	from RSU	√	n/a	n/a	×	√	√	n/a	√
	from NSP	×	√	n/a	×	√	×	n/a	√
Location Privacy	from outsiders	√	×	√	√	√	√	√	√
	from RSU	×	n/a	n/a	√	√	√	√	√
	from NSP	×	×	×	√	√	×	√	√
Start Point and Destination Privacy	from outsiders	√	×	√	√	√	√	√	√
	from RSU	×	n/a	n/a	√	√	√	√	√
	from NSP	×	×	×	√	×	×	√	√
Route Privacy	from outsiders	√	n/a	√	n/a	√	√	√	√
	from RSU	×	n/a	n/a	n/a	√	√	√	√
	from NSP	×	n/a	×	n/a	×	×	√	√
Unlinkability	identities	×	×	×	×	×	√	n/a	√
	routes	×	×	√	n/a	√	√	√	√
	identity and route	×	×	×	×	√	√	n/a	√
Authentication	authentication	√	√	×	√	√	√	√	√

Zhang et al. [23] presented a location privacy preserving mechanism for navigation services on smartphones, named ShiftRoute. The scheme shifts the destinations of route queries to the ones close-by to make mobile users to query navigation services for routes, without exposing sensitive location information. But the ShiftRoute did not consider the identity privacy and preserve the location and route privacy.

Ni et al. [24] later put forward PrivAV, which is a privacy-preserving valet parking in autonomous driving era. PrivAV realizes the anonymous service authentication between users and automated valet parking servers and preserves the location privacy preservation for users. Same as P-SPAN, PrivAV still reveals vehicle's parking lot to the automated valet parking servers. And it didn't consider the unlinkability between the user and the vehicle.

To further protect vehicle user's privacy, Li et al. [25] introduced a novel privacy-preserving navigation scheme, called PiSim. PiSim supports similar queries in navigation services, and it focuses on preserving the locations and routes privacy of vehicle users under the scenario of same start point and destinations queries to service provider. In this scheme, they trans-form the typical navigation approach into a traffic congestion querying approach, which may affect the timeliness and efficiency of the navigation service.

Zhong et al. [26] designed a pseudonym management mechanism based on vehicle movement regularity to protect the location privacy of vehicles in vehicular networks. But in the scheme, each vehicle needs to be authorized a large number of pseudonyms and certificates by TA, which is a heavy workload.

Baruah et al. [27] recently put forward a security and privacy preserved vehicle navigation system, aiming to retrieve the navigation result to guide the vehicle users through the optimal routes towards their destination. Different with other systems, their proposed navigation system does not disclose the destination information not only to the outside attackers but also to RSUs and trusted authority. Nevertheless, the proposed navigation system does not consider the identity privacy preserving.

According to the above listed related works in recent year, there still remain some privacy issues in navigation system for vehicular networks to be addressed. Thus, towards these issues, we design a novel dual privacy-preserving lightweight navigation system for vehicular networks in this paper. TABLE 1 provides the comparison of multi-dimension privacy properties between our proposed navigation system and the existing system.

III. PRELIMINARIES

Here we review the related assumptions in elliptic curve cryptography and key derivation function (KDF) employed in the proposed scheme.

A. ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

Let q denote a large prime integer, G be a cyclic group with generator (base point) P, and its order is q.

Definition 1 (Elliptic Curve Discrete Logarithm Problem (ECDLP) [28]): Given two points $X, Y \in G$, finding a $\alpha \in Z_q^*$ such that $Y = \alpha X$ is computationally infeasible. Let $Adv_{ECDLP}(\mathcal{A})$ denote the probability that $\mathcal{A}$ can solve ECDLP. The ECDLP assumption is that for any probabilistic polynomial time (PPT)-bounded adversary $\mathcal{A}$, $Adv_{ECDLP}(\mathcal{A})$ is negligible, that is $Adv_{ECDLP}(\mathcal{A}) < \varepsilon$, for sufficiently small ε.

Definition 2 (Elliptic Curve Diffie-Hellman Problem (ECDHP) [28]): Given two points $X = xP$ and $Y = yP$, where $x, y \in Z_q^*$ and $X, Y \in G$, finding a point $Z = xyP$ is computationally infeasible. Let $Adv_{ECDHP}(\mathcal{A})$ denote the probability that $\mathcal{A}$ can solve ECDHP. The ECDHP assumption is that for any PPT-bounded adversary $\mathcal{A}$, $Adv_{ECDHP}(\mathcal{A})$ is negligible, that is $Adv_{ECDHP}(\mathcal{A}) < \varepsilon$, for sufficiently small ε.

Definition 3 (Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP) [28]): Given three points $X = xP$, $Y = yP$ and $Z = zP$, where $x, y, z \in Z_q^*$ and $X, Y, Z \in G$, deciding $xy \equiv z$ is computationally infeasible. Let $Adv_{ECDDHP}(\mathcal{A})$ denote the probability that $\mathcal{A}$ can solve ECDDHP. The ECDDHP assumption is that for any PPT-bounded adversary $\mathcal{A}$, $Adv_{ECDDHP}(\mathcal{A})$ is negligible, that is $Adv_{ECDDHP}(\mathcal{A}) < \varepsilon$, for sufficiently small ε.

B. KEY DERIVATION FUNCTION (KDF)

As the basic and essential component of cryptographic systems, the key derivation function (KDF) aims to take some source of initial keying material and derive from it one or more cryptographically strong secret keys [29]. The definition of KDF is as follows.

Definition 4 (A Key Derivation Function (KDF) [29]): Taking four parameters as input, which includes a value σ sampled from a source of keying material, a length value ℓ, a salt value γ defined over a set of possible salt values, and a context variable c. The last parameters are optional, that is, they can be set to the null string or to a constant. The KDF output is a string of length ℓ. A KDF is modeled as a random oracle.

IV. OVERVIEW

This section first introduces the system model, and then describes the threat model and security model.

A. SYSTEM MODEL

The system model of the proposed navigation system consists of Trusted Authority (TA), Navigation Service Provider (NSP), On-Board Unit (OBU) and Roadside Unit (RSU) as shown in Fig.1. An OBU relies on the GNSS and ground receivers to know its current location. The details are as follows.

- **Trusted Authority (TA):** TA is the trusted authority, which is responsible for the system initialization and registration of NSP, RSUs and OBUs of vehicles.
- **Navigation Service Provider (NSP):** NSP is the central navigation service provider. It analyzes the real-time traffic information collected by RSUs and provides optimal routes for vehicles according to the start points and destinations queried by OBUs to guide vehicles into the road with low traffic jam, so as to support comfortable driving.

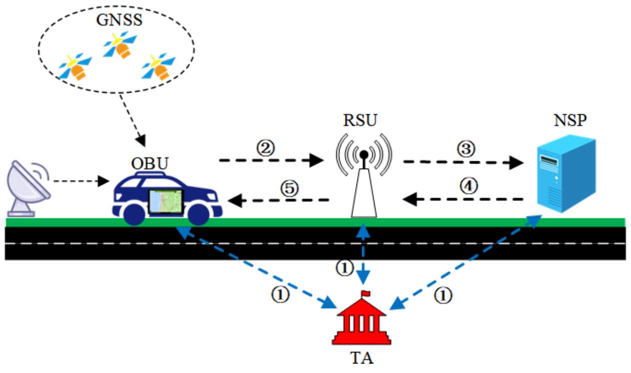

FIGURE 1. System model.

- **On-Board Unit (OBU):** OBUs are installed on vehicles. OBU is in charge of requesting and receiving navigation service independently according to the start point and destination of its query. In order to improve the accuracy of its location, OBU utilizes GNSS information and vehicle sensors information to estimate its location and velocity. Then OBU sends the route query to NSP through its nearby RSU. After verification, it receives the real-time optimal driving route from NSP. During the process of navigation service provision, the vehicle maintains anonymity and its route privacy.
- **Roadside Unit (RSU):** RSU is mounted on the roadside. It is a communication device, which is responsible for verifying navigation queries of OBUs passing by and transmitting queries to NSP. Besides, it also collects real-time traffic information simultaneously to assist NSP in determining the optimal route with low congestion.

We now introduce the process of the proposed navigation system according to the Fig.1.

① TA firstly initializes the whole system. And then it generates the public-private key pair for OBU, RSU and NSP to register them.

② OBU of a vehicle anonymously and unlinkablely submits sign-encrypted navigation query to its nearby RSU.

③ The RSU verifies its signature of the query. If it is valid, the RSU will sign and transmit the query to NSP. Otherwise, it will abort the query request.

④ NSP verifies the signature of RSU and decrypts the navigation query to obtain the start point and destination. Then it encrypts the current optimal route and sends to the RSU.

⑤ RSU transmits the encrypted route information to the OBU. And OBU can utilize the same symmetric key to get the optimal driving route.

B. THREAT MODEL

- TA is trusted to generate the public-private key pairs of all registrants and keep their private keys secretly.

- OBU is trusted to protect the vehicle's private key and to perform cryptographic operations. We assume no side-channel attacks and operations within the OBU are secure.
- RSU and NSP are honest-but-curious [30]. RSUs can collude with each other and try to link the queries to a same vehicle. NSP wants to link a route belonging to a specific vehicle. They provide the correct safety services to the vehicle but want to track and link vehicles based on the messages.

The communications between TA and other entities are secure.

C. SECURITY MODEL

This subsection defines the security models for the designed signature scheme (**Algorithm 1** and **Algorithm 2** in the proposed navigation system), which will be used to prove the security features including unforgeability and anonymity of the designed signature scheme in Section 6.2. We first define the following oracles to simulate the abilities of the adversary in breaking the security of the signature scheme adopted in the proposed system:

- Joining oracle $\mathcal{O_J}$: Taking the public parameters *Param* as input, $\mathcal{O_J}$ returns the new OBU's private-public key pair (sk_{OBU}, pk_{OBU}) of the new user, where $sk_{OBU} \in SK$ and $pk_{OBU} \in PK$.
- Secret parameter generation Oracle $\mathcal{O_{SP}}$: Taking the *Param* and *sk* of the system as input, $\mathcal{O_{SP}}$ returns the secret parameter sp_{OBU}.
- Signing Oracle $\mathcal{O_S}$: The oracle takes the input of the public parameters *Param*, an event identifier E, the number of public keys n, a set Y contains n public keys $\{pk_1, pk_2, \ldots, pk_i, \ldots, pk_n\}$ where $Y \subseteq PK$ and $i \in [1, n]$, a ring key rk_{OBU} and a message M, outputs a signature σ.

1) UNFORGEABILITY GAME

The unforgeability game is played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. During the game, $\mathcal{A}$ can query $\mathcal{O_J}$, $\mathcal{O_{SP}}$, $\mathcal{O_S}$. Set that q_j is the maximum number of $\mathcal{O_J}$ queries, q_p is the maximum number of $\mathcal{O_{SP}}$ queries, and q_s is the maximum number of $\mathcal{O_S}$ queries.

- *Setup*: This algorithm takes as input a security parameter λ, and outputs *Param* to $\mathcal{A}$.
- *Queries* : $\mathcal{A}$ makes queries to $\mathcal{O_J}$, $\mathcal{O_{SP}}$, $\mathcal{O_S}$ for polynomial many times.
- *Forgery* : $\mathcal{A}$ forges $(E_\mathcal{C}, M_\mathcal{C}, \sigma_\mathcal{C})$ for $Y_\mathcal{C}$ of n public keys.

If the following conditions are satisfied, $\mathcal{A}$ will win the game:

a. $SigVerify(Param, E_\mathcal{C}, n, Y_\mathcal{C}, \sigma_\mathcal{C}, M_\mathcal{C}) = 1$.
b. All public keys in $Y_\mathcal{C}$ are query outputs of $\mathcal{O_J}$.
c. $Y_\mathcal{C}, M_\mathcal{C}, E_\mathcal{C}$ are not query outputs of $\mathcal{O_S}$.

Let $Adv_{\mathcal{A}}^{Unforge}(\lambda, q_j, q_p, q_s) = \Pr[\mathcal{A}\ wins\ the\ game]$ denote the probability of the PPT adversary $\mathcal{A}$ winning the above unforgeability game.

Definition 5: Unforgeability: The designed signature scheme in our proposed system satisfies unforgeability if $Adv_{\mathcal{A}}^{Unforge}(\lambda, q_j, q_p, q_s)$ is negligible.

2) ANONYMITY GAME

The probability of any verifier correctly guessing the signer's identity in signature scheme should not be greater than $1/n$ when the owners of public keys are unknown. What's more, due to that TA knows all the information and keys of vehicles in our proposed system, anyone who has the TA's private key can break the anonymity. That is, if the private key of the TA in our proposed system has not been obtained, the scheme satisfies anonymity. The anonymity game is also played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. During the game, $\mathcal{A}$ can query $\mathcal{O_J}$, $\mathcal{O_S}$, $\mathcal{O_S}$. Set that q_j is the maximum number of $\mathcal{O_J}$ queries, q_p is the maximum number of $\mathcal{O_S}$ queries, q_s is the maximum number of $\mathcal{O_S}$ queries.

- *Setup*: This algorithm takes a security parameter λ as input and returns *Param* to $\mathcal{A}$.
- *Queries*: $\mathcal{A}$ makes queries to $\mathcal{O_J}$, $\mathcal{O_{SP}}$ and $\mathcal{O_S}$ for polynomial many times.
- *Chellenge*: $\mathcal{A}$ gives $\mathcal{C}$ its forgeries including a set $Y_\mathcal{C}$. Note that all public keys in $Y_\mathcal{C}$ are query outputs of $\mathcal{O_J}$, a message $M_\mathcal{C}$ and an event identifier $E_\mathcal{C}$. $\mathcal{C}$ randomly picks $\pi \in [1, n]$ and utilize sk_π to generate a challenge signature σ_π, where sk_π is a corresponding private key of $pk_\pi \in Y_\mathcal{C}$. Then σ_π is returned to $\mathcal{A}$.
- *Guess*: $\mathcal{A}$ guesses $\pi_\mathcal{C} \in [1, n]$.

Let $Adv_{\mathcal{A}}^{Anony}(\lambda, q_j, q_p, q_s) = |\Pr[\pi_\mathcal{C} = \pi] - \frac{1}{n}|$ denote the probability of the PPT adversary $\mathcal{A}$ winning the above anonymity game.

Definition 6: Anonymity: The designed signature scheme in our proposed system satisfies anonymity if $Adv_{\mathcal{A}}^{Anony}(\lambda, q_j, q_p, q_s)$ is negligible.

V. THE PROPOSED NAVIGATION SYSTEM

The proposed navigation system includes three phases, named *System Initialization*, *Registration* and *Navigation Service Provision*. The detailed description of each phase is in the following.

A. SYSTEM INITIALIZATION PHASE

To initialize the navigation system, firstly, TA inputs security parameter 1^κ to generate a cyclic group G with base point P and prime order q. And then it initializes three cryptographic hash functions including $H_1 : G \to G$, $H_2 : G \times G \times G \times G \to Z_q^*$ and $H_3 : G \times G \times G \times G \times \{0, 1\}^* \to Z_q^*$, and a KDF. After that, TA randomly selects $sk_{TA} \in Z_q^*$ as its private key and computes its corresponding public key $pk_{TA} = sk_{TA} \cdot P$. Finally, TA publishes the public parameters $Param = \{G, P, q, H_1, H_2, H_3, pk_{TA}\}$.

B. REGISTRATION PHASE

This phase presents the registration of NSP, RSUs and OBUs.

1) REGISTRATION OF NSP

Before NSP providing navigation service, it needs to register with TA. Firstly, NSP submits its registration information including service type, identity number, phone number etc. to TA for registration. TA waits for out-of-band documents to determine the authenticity of the registration information. After that, TA generates key pair (sk_{NSP}, pk_{NSP}) for NSP, where $pk_{NSP} = sk_{NSP} \cdot P$ is its public key and $sk_{NSP} \in Z_q^*$. TA returns the public-private key pair to NSP through secure channel.

2) REGISTRATION OF RSU

RSUs also need to register with TA, taking RSU_i as an example. Firstly, RSU_i submits its registration information including location, identity number etc. to TA for registration. After that, TA generates key pair (sk_{RSU_i}, pk_{RSU_i}) for RSU_i, where $pk_{RSU_i} = sk_{RSU_i} \cdot P$ is its public key and $sk_{RSU_i} \in Z_q^*$. TA returns the public-private key pair to RSU_i through secure channel.

3) REGISTRATION OF OBU

If a vehicle OBU_i would like to join the navigation system, it needs to register with TA. Firstly, OBU_i submits its registration information including license plate number, identity number, phone number etc. to TA for registration. TA waits for out-of-band documents to determine the authenticity of the registration information. After that, TA generates key pair (sk_{OBU_i}, pk_{OBU_i}) for OBU_i, where $pk_{OBU_i} = sk_{OBU_i} \cdot P$ is its public key and $sk_{OBU_i} \in Z_q^*$. In addition, TA randomly selects $r \in Z_q^*$, and then it computes $R_1 = r \cdot P$, $h_1 = H_1(R_1)$, $x_1 = sk_{TA} \cdot h_1$ and $y_1 = r \cdot h_1$. In the following, it continues calculating $k_1 = H_2(R_1, x_1, y_1, pk_{TA})$ and $s = r + k_1 \times sk_{TA}$. Finally, the secret parameters are generated, that is $sp_{OBU_i} = (x_1, k_1, s)$. And it stores R_1 and sp_{OBU_i}. Finally, TA returns the key pair and secret parameters $(sk_{OBU_i}, pk_{OBU_i}, sp_{OBU_i})$ to OBU_i through a secure channel.

When OBU_i receives the key pair and secret parameters, it will first utilize the public key pk_{TA} of TA to compute $R_1' = s \cdot P - k_1 \cdot pk_{TA}$, $h_1' = H_1(R_1')$, $y_1' = s \cdot h_1' - k_1 \cdot x_1$ and $k_1' = H_2(R_1', x_1, y_1', pk_{TA})$. And then if $k_1' = k_1$, OBU_i will generate its ring key $rk_{OBU_i} = (t, k_1)$ where $t = s + k_1 \times sk_{OBU_i}$. Finally, it will keep its secret key sk_{OBU_i} and rk_{OBU_i} secretly.

C. NAVIGATION SERVICE PROVISION PHASE

This phase introduces the process of navigation service provision.

1) NAVIGATION QUERY OF OBU

When a vehicle OBU_i wants to request navigation service, it first randomly selects $a \in Z_q^*$ and computes $pka = a \cdot P$, $PK = a \cdot pk_{NSP}$. And then OBU_i calculates a symmetric key through $symk = KDF(PK)$. After that, OBU_i utilizes the symmetric key $symk$ to encrypt the start point $StaPt$ and destination $Dest$ that it wants to query, and gets the ciphertext $C = Enc_{symk}(StaPt, Dest)$.

Next, OBU_i utilizes n valid public keys $\{pk_1, pk_2, \ldots, pk_i, \ldots, pk_n\}$ where $pk_i = pk_{OBU_i}$ that is OBU_i own public key to sign the ciphertext C calculated above. In this step, it uses $Param$, an event identifier E, n valid public keys $\{pk_1, pk_2, \ldots, pk_i, \ldots, pk_n\}$, and rk_{OBU_i} to perform **Algorithm 1**. Through **Algorithm 1**, OBU_i can generate the signature for its encrypted start point, destination and hide its public key in n public keys. **Algorithm 1** is shown as following.

Algorithm 1 : *SigGen*

```
Input: $Param, E, n, \{pk_1, pk_2, \ldots, pk_i, \ldots, pk_n\}, rk_{OBU_i}, C, pka$
Output: A signature on $C$ and $pka$
1: Randomly select $u, v \leftarrow Z_q^*$
2: $R_2 = u \cdot P + \sum_{l=1}^{n} pk_l$
3: $h_2 = H_1(R_2)$
4: $x_2 = t \times v \cdot h_2$
5: $y_2 = u \cdot h_2$
6: $K = k_1 \times v$
7: $k_2 = H_3(R_2, x_2, y_2, pk_{NSP}, E, K, C, pka)$
8: $T = u + k_2 \times v \times t$
9: $R_1'' = k_2 \times v \cdot (R_1' + k_1 \cdot pk_{OBU_i})$
10: Return $Sig = (x_2, K, k_2, T, R_1'')$
```

Finally, OBU_i sends $(sig, C, E, n, \{pk_1, \ldots, pk_n\}, pka)$ to it nearby RSU, for example, RSU_j.

2) SIGNATURE VERIFICATION BY RSU

After receiving the signature related information $(sig, C, E, n, \{pk_1, \ldots, pk_n\}, pka)$ from OBU_i, RSU_j performs **Algorithm 2** to verify the validity of the signature. **Algorithm 2** is shown below.

Algorithm 2 *SigVerify*

```
Input: $Param, E, n, \{pk_1, pk_2, \ldots, pk_i, \ldots, pk_n\}, Sig, C, pka$
Output: The verification result of the message-signature pair
1: $R_2' = T \cdot P - R_1'' - K \times k_2 \cdot pk_{CCS} + \sum_{l=1}^{n} pk_l$
2: $h_2' = H_1(R_2')$
3: $y_2' = T \cdot h_2' - k_2 \cdot x_2$
4: $k_2' = H_3(R_2', x_2, y_2', pk_{CCS}, E, K, C, pka)$
5: If $k_2' = k_2$
6:     Return TRUE
7: Else
8:     Return FALSE
```

If the algorithm returns TRUE, RSU_j will use its private key to sign the information including (C, pka) through the elliptic curve digital signature algorithm (ECDSA) [31]. And then it sends the signed (C, pka) to NSP. Else, it will drop the message.

3) NAVIGATION SERVICE PROVISION BY NSP

When NSP receives the signed (C, pka) from RSU, it first utilizes RSU_j's public key to verify the signature. If the signature is valid, NSP will use its private key sk_{NSP} to compute $PK' = sk_{NSP} \cdot pka$ and then generate a symmetric key through $symk' = KDF(PK')$. After that, it utilizes the symmetric key $symk'$ to decrypt the ciphertext C to get the start point $StaPt$ and destination $Dest$. Then NSP analyzes the real-time traffic information collected by RSUs to determine the optimal route RT with minimum driving time and congestion. Next, it encrypts the RT using the symmetric key $symk'$ to get the ciphertext $C' = Enc_{symk'}(RT)$. Finally, NSP transmits the ciphertext C' to RSU_j.

RSU_j forwards C' it received to OBU_i.

OBU_i can utilize the symmetric key $symk$ to decrypt C' to get the optimal route RT.

VI. CORRECTNESS AND SECURITY ANALYSIS

We first analyze the correctness of the proposed system, and then give the security proof of the designed signature scheme adopted in the proposed system. Finally, we introduce the privacy analysis of the proposed navigation system in this section.

A. CORRECTNESS ANALYSIS

1) THE CORRECTNESS OF VERIFICATION

Firstly, OBU_i needs to verify $k_1' = k_1$ in registration phase. Due to $k_1 = H_2(R_1, x_1, y_1, pk_{TA})$ and $k_1' = H_2(R_1', x_1, y_1', pk_{TA})$, if $k_1' = k_1$, it just needs to verify $R_1' = R_1$ and $y_1' = y_1$. In addition, the signature needs to be verified in anonymous charging phase. If the signature verification is valid, it means that $k_2' = k_2$ is true. According to $k_2' = H_3(R_2', x_2, y_2', pk_{NSP}, E, K, C, pka)$ and $k_2 = H_2(R_2, x_2, y_2, pk_{TA}, E, K, C, pka)$, we need to prove $R_2' = R_2$ and $y_2' = y_2$.

The proof of $R_1' = R_1$ and $y_1' = y_1$ is as following.

According to $s = r + k_1 \times sk_{TA}$,

$$\begin{aligned} R_1' &= s \cdot P - k_1 \cdot pk_{TA} \\ &= (r + k_1 \times sk_{TA}) \cdot P - k_1 \cdot pk_{TA} \\ &= r \cdot P + k_1 \times sk_{TA} \cdot P - k_1 \cdot pk_{TA} \\ &= r \cdot P + k_1 \cdot pk_{TA} - k_1 \cdot pk_{TA} \\ &= r \cdot P = R_1 \end{aligned}$$

Due to $R_1' = R_1$, $h_1' = H_1(R_1')$ and $h_1 = H_1(R_1)$, we can get $h_1' = h_1$. Then according to $s = r + k_1 \times sk_{TA}$ and $x_1 = sk_{TA} \cdot h_1$,

$$\begin{aligned} y_1' &= s \cdot h_1' - k_1 \cdot x_1 \\ &= (r + k_1 \times sk_{TA}) \cdot h_1 - k_1 \cdot (sk_{TA} \cdot h_1) \\ &= r \cdot h_1 + k_1 \times sk_{TA} \cdot h_1 - k_1 \times sk_{TA} \cdot h_1 \\ &= r \cdot h_1 = y_1 \end{aligned}$$

Thus, $R_1' = R_1$ and $y_1' = y_1$ are proved, and then thus $k_1' = k_1$.

The proof of $R_2' = R_2$ and $y_1' = y_1$ is as following.

According to $T = u + k_2 \times v \times t$, $t = s + k_1 \times sk_{OBU_i}$, $s = r + k_1 \times sk_{TA}$, $R_1'' = k_2 \times v \cdot (R_1' + k_1 \cdot pk_{OBU_i})$, and $R_1' = r \cdot P$, $K = k_1 \times v$, then

$$\begin{aligned} R_2' &= T \cdot P - R_1'' - K \times k_2 \cdot pk_{TA} + \sum_{l=1}^{n} pk_l \\ &= (u + k_2 \times v \times t) \cdot P - k_2 \times v \cdot (R_1' + k_1 \cdot pk_{OBU_i}) \\ &\quad - K \times k_2 \cdot pk_{TA} + \sum_{l=1}^{n} pk_l \\ &= (u + k_2 \times v \times (s + k_1 \cdot sk_{OBU_i})) \cdot P - k_2 \\ &\quad \times v \cdot (r \cdot P + k_1 \cdot pk_{OBU_i}) \\ &\quad - k_1 \times v \times k_2 \cdot pk_{TA} + \sum_{l=1}^{n} pk_l \\ &= (u + k_2 \times v \times (r + k_1 \times sk_{TA} + k_1 \times sk_{OBU_i})) \\ &\quad \cdot P - k_2 \times v \cdot (r \cdot P \\ &\quad + k_1 \cdot pk_{OBU_i}) - k_1 \times v \times k_2 \cdot pk_{TA} + \sum_{l=1}^{n} pk_l \\ &= u \cdot P + k_2 \times v \times r \cdot P + k_2 \\ &\quad \times v \times k_1 \cdot pk_{TA} + k_2 \times v \times k_1 \cdot pk_{OBU_i} \\ &\quad - k_2 \times v \times r \cdot P - k_2 \\ &\quad \times v \times k_1 \cdot pk_{OBU_i} - k_1 \times v \times k_2 \cdot pk_{TA} + \sum_{l=1}^{n} pk_l \\ &= u \cdot P + \sum_{l=1}^{n} pk_l \\ &= R_2 \end{aligned}$$

Due to $R_2' = R_2$, $h_2' = H_1(R_2')$ and $h_2 = H_1(R_2)$, we can get $h_2' = h_2$. Then according to $T = u + k_2 \times v \times t$, $t = s + k_1 \times sk_{OBU_i}$, $s = r + k_1 \times sk_{TA}$, and $x_2 = t \times v \cdot h_2$,

$$\begin{aligned} y_2' &= T \cdot h_2' - k_2 \cdot x_2 \\ &= (u + k_2 \times v \times t) \cdot h_2 - k_2 \cdot (t \times v \cdot h_2) \\ &= u \cdot h_2 \\ &= y_2 \end{aligned}$$

Thus, $R_2' = R_2$ and $y_2' = y_2$ are proved, and then thus $k_2' = k_2$. So far, the correctness of verification has been proved.

2) THE CORRECTNESS OF ENCRYPTION AND DECRYPTION

As stated in navigation service provision phase, OBU_i and NSP utilize their own calculated symmetric key to execute encryption and decryption operations. Here we prove that the symmetric keys $symk$ and $symk'$ calculated by them are the same.

$$\begin{aligned} symk &= KDF(PK) & symk' &= KDF(PK') \\ &= KDF(a \cdot pk_{NSP}) \quad \text{and} & &= KDF(sk_{NSP} \cdot pka) \\ &= KDF(a \times sk_{NSP} \cdot P) & &= KDF(sk_{NSP} \times a \cdot P) \end{aligned}$$

Therefore, $symk = symk'$. The correctness of encryption and decryption is proved.

B. SECURITY ANALYSIS

Theorem 1 (Unforgeability): The designed signature scheme used in our proposed system satisfies unforgeability in the random oracle model under the ECDHP assumption.

Proof: Let $\mathcal{A}$ be an adversary attacking the designed signature scheme used in our proposed system. Let $\mathcal{S}$ be a simulator. Assume $\mathcal{S}$ can solve ECDHP with probability $Adv_{\mathcal{A}}^{Unforge}(\lambda, q_j, q_p, q_s)$. Given $\{P, xP, yP\} \in G$, $\mathcal{S}$ can utilize $\mathcal{A}$ to compute xyP. The simulation is as following.

Setup: The simulator $\mathcal{S}$ takes a security parameter 1^{κ} as input. G is a cyclic group with generator P and prime order q. Also, it simulates $H_1 : G \to G, H_2 : G \times G \times G \times G \to Z_q^*$ and $H_3 : G \times G \times G \times G \times \{0, 1\}^* \to Z_q^*$ through random oracles H_1, H_2 and H_3. Then, assuming that $\mathcal{C}$ is a challenger with private key $sk_{\mathcal{C}}$ and public key $pk_{\mathcal{C}} = sk_{\mathcal{C}} \cdot P$. The simulator $\mathcal{S}$ does not know its private key $sk_{\mathcal{C}} = x$. Besides, $\mathcal{S}$ does not know the value of y, nor does it know that yP is used to be the output of H_1 queries. After that, $\mathcal{S}$ publishes the public parameters $Param = \{G, P, q\}$. In addition, $\mathcal{S}$ chooses $sk_{\mathcal{S}} \in Z_q^*$ randomly as the master private key of the system. Then it uses the master private key to compute public key $pk_{\mathcal{S}} = sk_{\mathcal{S}} \cdot P$, and publishes $pk_{\mathcal{S}}$.

Queries: The simulator $\mathcal{S}$ simulates the following oracles to answer the queries of adversary $\mathcal{A}$.

- H_1 queries: If it has not been queried, $\mathcal{S}$ randomly chooses $a \in Z_q^*$, computes and returns $a(yP) = ayP$ to $\mathcal{A}$. Or, $\mathcal{S}$ returns the output of the same query. At the same time, $\mathcal{S}$ stores (a, ayP) to $LIST_1$.
- H_2 queries: If it has not been queried, $\mathcal{S}$ returns a new result to $\mathcal{A}$. Or, $\mathcal{S}$ returns the output of the same query.
- H_3 queries: If it has not been queried, $\mathcal{S}$ returns a new result to $\mathcal{A}$. Or, $\mathcal{S}$ returns the output of the same query.
- $\mathcal{O}_{\mathcal{J}}$ queries: If it has not been queried, $\mathcal{S}$ returns a new private-public key pair $(sk_{\mathcal{A}}, pk_{\mathcal{A}})$ to $\mathcal{A}$. Or, $\mathcal{S}$ returns the same result. At the same time, $\mathcal{S}$ stores $(sk_{\mathcal{A}}, pk_{\mathcal{A}})$ to $LIST_2$.
- $\mathcal{O}_{\mathcal{SP}}$ queries: Inputting $Param$ and $pk_{\mathcal{S}}$, $\mathcal{S}$ randomly chooses $d \in Z_q^*$, calculates $R_1 = d \cdot P$, $h_1 = H_1(R_1)$, $x_1 = sk_{\mathcal{S}} \cdot h_1$, $y_1 = d \cdot h_1$, $k_1 = H_2(R_1, x_1, y_1, pk_{\mathcal{S}})$, $s = d + k_1 \times sk_{\mathcal{S}}$, and returns $sp_{\mathcal{A}} = (x_1, k_1, s)$ to $\mathcal{A}$.
- $\mathcal{O}_{\mathcal{S}}$ queries: Inputting $Param$, $pk_{\mathcal{S}}$, an event identifier E, the number of public keys n, a set Y where $pk_i \in Y$, and a message M, $\mathcal{S}$ performs the following operations:

a. $\mathcal{S}$ randomly chooses $d \in Z_q^*$, calculates $R_1 = d \cdot P$, $h_1 = H_1(R_1)$, $x_1 = sk_{\mathcal{S}} \cdot h_1$, $y_1 = d \cdot h_1$, $k_1 = H_2(R_1, x_1, y_1, pk_{\mathcal{S}})$.

b. $\mathcal{S}$ randomly chooses $k_2, u, v \in Z_q^*$, calculates $R_2 = u \cdot P - k_2 v \cdot (d \cdot P + k_1 sk_{\mathcal{S}} \cdot P) - k_1 k_2 v \cdot pk_i + \sum_{l=1}^{n} pk_l$, and queries H_1 for R_2. If it has been output, $\mathcal{S}$ aborts. Otherwise, $\mathcal{S}$ continues performing the following steps.

c. $\mathcal{S}$ randomly chooses $w \in Z_q^*$, calculates $x_2 = wv \times (d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 \cdot pk_i) = wv(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 sk_i \cdot P)$. Let $h_2 = H_1(R_2) = w \cdot P$, which satisfies $H_1(v^{-1} R_2) = H_1(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 sk_i \cdot P) = d + k_1 sk_{\mathcal{S}} + k_1 \times sk_i$.

d. $\mathcal{S}$ calculates $y_2 = u \cdot h_2 - k_2 \cdot h_2$, $R_1^{''} = k_2 \times v \cdot (R_1 + k_1 \cdot pk_i)$ and $K = k_1 \times v$. And then $\mathcal{S}$ queries H_3. If $(R_2, x_2, y_2, pk_{\mathcal{S}}, k_1 v, E, M)$ has been queried, $\mathcal{S}$ aborts. Or, $\mathcal{S}$ continues performing the following steps.

e. $\mathcal{S}$ returns a signature $sig = \{R_1^{''}, K, k_2, x_2, T\}$ to $\mathcal{A}$, and stores the tuple (d, v, k_1) to $LIST_3$.

Forgery: If $\mathcal{S}$ does not abort the above the queries, $\mathcal{A}$ will return a forgery $(E_{\mathcal{C}}, M_{\mathcal{C}}, \sigma_{\mathcal{C}}, Y_{\mathcal{C}})$ for $\mathcal{C}$ with probability at least ε, where $pk_{\mathcal{C}} \in Y_{\mathcal{C}}$. And the forgery satisfies:

a. $SigVerify(Param, E_{\mathcal{C}}n, Y_{\mathcal{C}}, sig_{\mathcal{C}}, M_{\mathcal{C}}) = 1$.
b. All public keys in $Y_{\mathcal{C}}$ are query outputs of $\mathcal{O}_{\mathcal{J}}$.
c. $Y_{\mathcal{C}}, M_{\mathcal{C}}, E_{\mathcal{C}}$ are not query outputs of $\mathcal{O}_{\mathcal{S}}$.

$\mathcal{S}$ will abort if H_1 queries are not required by $\mathcal{A}$ or one of $LIST_1$, $LIST_2$, $LIST_3$ is empty. Otherwise, $\mathcal{S}$ can get $h_2 = H_1(*) = ayP$. If $H_1(v^{-1}R_2) = H_1(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 sk_{\mathcal{C}} \cdot P) = d + k_1 sk_{\mathcal{S}} + k_1 sk_{\mathcal{C}}$, x_2 can be written as $x_2 = ayv(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 \cdot pk_{\mathcal{C}}) = ayv(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 sk_{\mathcal{C}} \cdot P) = ayvd \cdot P + ayvk_1 sk_{\mathcal{S}} \cdot P + ayvk_1 sk_{\mathcal{C}} \cdot P = vd \cdot h_2 + vk_1 sk_{\mathcal{S}} \cdot h_2 + avk_1(xy \cdot P)$. After that, $\mathcal{S}$ can compute $(avk_1)^{-1}(x_2 - (vd + vk_1 sk_{\mathcal{S}}) \cdot h_2) = xyP$, which is the solution of the given ECDHP.

If $\mathcal{S}$ does not abort during the whole simulation process, it needs to not abort all $\mathcal{O}_{\mathcal{S}}$ queries. Firstly, $\mathcal{S}$ queries H_1 for R_2. Due to $d, k_2, u, v \in Z_q^*$, they are uniformly distributed in Z_q^4, H_1's collision probability is $\frac{q_h}{2^{4q}}$. Thus, the probability of fail queries is at most $\frac{q_h q_s}{2^{4q}}$. And $\mathcal{S}$ queries H_3 for $(R_2, x_2, y_2, pk_{\mathcal{S}}, k_1 v, E, M)$. w is uniformly distributed in Z_q according to $w \in Z_q^*$, thus H_3's collision probability is $\frac{q_h + q_s}{2^q}$. Finally, we can get that the probability of fail queries is at most $\frac{q_s(q_h+q_s)}{2^q}$.

In addition, let $event_1$ denote that $\mathcal{A}$ forged a valid signature without querying H_1 queries, $event_2$ denote that $\mathcal{A}$ forged a valid signature with $H_1(v^{-1}R_2) \neq H_1(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 sk_i \cdot P)$. If $event_1 \cup event_2$ occurs, ECDHP cannot be solved. The probability of $event_1 \cup event_2$ occurrence is $\Pr[event_1 \cup event_2] = \Pr[event_1 \cap \neg event_2] + \Pr[event_2]$. From $event_1$ and $\neg event_2$, we can get that $[v(d + k_1 sk_{\mathcal{S}} + k_1 x)]^{-1} x_2 = h_2 = H_1(R_2)$. Due to the selection of all variants in Z_q^*, $\Pr[event_1 \cup event_2]$ is at most $\frac{1}{2^q}$. About $event_2$, let $R_2 = U \cdot P + \sum_{l=1}^{n} pk_l$, $h_2 = H_1(R_2)$, $y_2 = U' \cdot h_2$, $x_2 = t'_{\mathcal{C}} v' \cdot h_2 \neq t_{\mathcal{C}} v \cdot h_2$. Since $sig = \{R_1^{''}, K, k_2, x_2, T\}$ is valid, we can get that $R_2 = u \cdot P - t_{\mathcal{C}} v k_2 \cdot P + \sum_{l=1}^{n} pk_l$, $h_2 = H_1(R_2)$, $y_2 = u \cdot h_2 - k_2 \cdot x_2$, $U = u - k_2 t_{\mathcal{C}} v$, $U' = u - k_2 t'_{\mathcal{C}} v'$. Thus, $k_2 = \frac{U' - U}{t_{\mathcal{C}} v - t'_{\mathcal{C}} v'} = H_3(*)$. Due to the distribution of H_3 queries in Z_q, the probability of querying H_3 for k_2 is at most $\frac{q_h + q_s}{2^q}$.

Finally, we can get the probability of $\mathcal{S}$ computing the solution of the given ECDHP from the forgery is at least $Adv_{\mathcal{A}}^{Unforge}(\lambda, q_j, q_p, q_s) = \varepsilon - \frac{q_h q_s}{2^{4q}} - \frac{q_s(q_h+q_s)}{2^q} - \frac{1}{2^q} - \frac{q_h+q_s}{2^q} < \varepsilon$.

Based on the above arguments, **Theorem 1** holds. ■

Theorem 2 (Anonymity): The designed signature scheme used in our proposed system satisfies anonymity in the random oracle model under the ECDDHP assumption.

Proof: Let $\mathcal{A}$ be an adversary attacking the designed signature scheme used in our proposed system. Let $\mathcal{S}$ be a simulator. Assume $\mathcal{S}$ can solve ECDDHP with probability $Adv_{\mathcal{A}}^{Anony}(\lambda, q_j, q_p, q_s)$. Given $\{P, x_0P, yP, x_1yP\} \in G$, $\mathcal{S}$ can utilize $\mathcal{A}$ to compute $x_1y = x_0y$. The simulation is as following.

Setup: The simulator $\mathcal{S}$ takes as input a security parameter 1^{κ}. G is a cyclic group with generator P and prime order q. Also, it simulates $H_1 : G \to G, H_2 : G \times G \times G \times G \to Z_q^*$ and $H_3 : G \times G \times G \times G \times \{0,1\}^* \to Z_q^*$ through random oracles H_1, H_2 and H_3. Then, assuming that $\mathcal{C}_0, \mathcal{C}_1$ are two challengers. Their private keys are $sk_{\mathcal{C}_0}$ and $sk_{\mathcal{C}_1}$ respectively, and their corresponding public keys are $pk_{\mathcal{C}_0} = sk_{\mathcal{C}_0} \cdot P$ and $pk_{\mathcal{C}_1} = sk_{\mathcal{C}_1} \cdot P$ respectively. The simulator $\mathcal{S}$ does not know their private keys $sk_{\mathcal{C}_0} = x_0$ and $sk_{\mathcal{C}_1} = x_1$. $\mathcal{S}$ also does not know y, nor does it know that yP is used to be the output of H_1 queries. $\mathcal{S}$ publishes the public parameters $Param = \{G, P, q\}$. In addition, $\mathcal{S}$ chooses $sk_{\mathcal{S}} \in Z_q^*$ randomly as the master private key of the system and uses the master private key to compute public key $pk_{\mathcal{S}} = sk_{\mathcal{S}} \cdot P$. Then it publishes $pk_{\mathcal{S}}$.

Queries 1: The simulator $\mathcal{S}$ simulates the following oracles to answer the queries of adversary $\mathcal{A}$.

- H_1 queries: If it has not been queried, $\mathcal{S}$ randomly chooses $a \in Z_q^*$, computes and returns $a(yP) = ayP$ to $\mathcal{A}$. Or, $\mathcal{S}$ returns the output of the same query. At the same time, $\mathcal{S}$ stores (a, ayP) to $LIST_1$.
- H_2 queries: If it has not been queried, $\mathcal{S}$ returns a new result to $\mathcal{A}$. Or, $\mathcal{S}$ returns the output of the same query.
- H_3 queries: If it has not been queried, $\mathcal{S}$ returns a new result to $\mathcal{A}$. Or, $\mathcal{S}$ returns the output of the same query.
- $\mathcal{O}_{\mathcal{J}}$ queries: If it has not been queried, $\mathcal{S}$ returns a new private-public key pair $(sk_{\mathcal{A}}, pk_{\mathcal{A}})$ to $\mathcal{A}$. Or, $\mathcal{S}$ returns the same result. At the same time, $\mathcal{S}$ stores $(sk_{\mathcal{A}}, pk_{\mathcal{A}})$ to $LIST_2$.
- $\mathcal{O}_{\mathcal{S}}$ queries: Inputting $Param$ and $pk_{\mathcal{S}}$, $\mathcal{S}$ randomly chooses $d \in Z_q^*$, calculates $R_1 = d \cdot P$, $h_1 = H_1(R_1)$, $x_1 = sk_{\mathcal{S}} \cdot h_1$, $y_1 = d \cdot h_1$, $k_1 = H_2(R_1, x_1, y_1, pk_{\mathcal{S}})$, $s = d + k_1 \times sk_{\mathcal{S}}$, and returns $sp_{\mathcal{A}} = (x_1, k_1, s)$ to $\mathcal{A}$.
- $\mathcal{O}_{\mathcal{S}}$ queries: Inputting $Param$, $pk_{\mathcal{S}}$, an event identifier E, the number of public keys n, a set Y where $pk_i \in Y$, and a message M, $\mathcal{S}$ performs the following operations:

a. $\mathcal{S}$ randomly chooses $d \in Z_q^*$, calculates $R_1 = d \cdot P$, $h_1 = H_1(R_1)$, $x_1 = sk_{\mathcal{S}} \cdot h_1$, $y_1 = d \cdot h_1$, $k_1 = H_2(R_1, x_1, y_1, pk_{\mathcal{S}})$.

b. $\mathcal{S}$ randomly chooses $k_2, u, v \in Z_q^*$, calculates $R_2 = u \cdot P - k_2 v \cdot (d \cdot P + k_1 sk_{\mathcal{S}} \cdot P) - k_1 k_2 v \cdot pk_i + \sum_{l=1}^{n} pk_l$, and queries H_1 for R_2. If it has been output, $\mathcal{S}$ aborts. Otherwise, $\mathcal{S}$ continues performing the following steps.

c. $\mathcal{S}$ randomly chooses $w \in Z_q^*$, calculates $x_2 = wv \times (d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 \cdot pk_i) = wv(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 \times sk_i \cdot P)$. Let $h_2 = H_1(R_2) = w \cdot P$, which satisfies $H_1(v^{-1}R_2) = H_1(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 sk_i \cdot P) = d + k_1 sk_{\mathcal{S}} + k_1 sk_i$.

d. $\mathcal{S}$ calculates $y_2 = u \cdot h_2 - k_2 \cdot h_2$, $R_1^{''} = k_2 \times v \cdot (R_1 + k_1 \cdot pk_i)$ and $K = k_1 \times v$. And then $\mathcal{S}$ queries H_3, if $(R_2, x_2, y_2, pk_{\mathcal{S}}, k_1 v, E, M)$ has been queried, $\mathcal{S}$ aborts. Or, $\mathcal{S}$ continues performing the following steps.

e. $\mathcal{S}$ returns a signature $sig = \{R_1^{''}, K, k_2, x_2, T\}$ to $\mathcal{A}$, and stores the tuple (d, v, k_1) to $LIST_3$.

Challenge: $\mathcal{A}$ returns the forgeries including $pk_{\mathcal{C}_0}$, $pk_{\mathcal{C}_1}$ and $(E_{\mathcal{C}}, M_{\mathcal{C}}, Y_{\mathcal{C}} \cup \{pk_{\mathcal{C}_0}\} \cup \{pk_{\mathcal{C}_1}\})$ for challengers. The forgeries satisfy that $pk_{\mathcal{C}_0}, pk_{\mathcal{C}_1}$ are not query outputs of $\mathcal{O}_{\mathcal{S}}$. The challengers randomly select a bit $b \in \{0,1\}$ and return $sig_{\mathcal{C}}$ to $\mathcal{A}$.

Queries 2: Repeat the steps of ***Queries 1***.

Guess: If $\mathcal{S}$ does not abort the above the queries, $\mathcal{A}$ will return a bit $b' \in \{0,1\}$ where $b' = b$, and a valid forgery $(E_{\mathcal{C}}, M_{\mathcal{C}}, \sigma_{\mathcal{C}}, Y_{\mathcal{C}})$ where $pk_{\mathcal{C}_{b'}} \in Y_{\mathcal{C}}$ for $\mathcal{C}_0$ and $\mathcal{C}_1$ with probability at least ε. $\mathcal{S}$ will abort if H_1 queries are not queried by $\mathcal{A}$ or one of $LIST_1, LIST_2, LIST_3$ is empty. Otherwise, $\mathcal{S}$ can get $h_2 = H_1(*) = ayP$. If $H_1(v^{-1}R_2) = H_1(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 sk_{\mathcal{C}_{b'}} \cdot P) = d + k_1 sk_{\mathcal{S}} + k_1 sk_{\mathcal{C}_{b'}}$, x_2 can be written as $x_2 = ayv(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 \cdot pk_{\mathcal{C}_{b'}}) = ayv(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 sk_{\mathcal{C}_{b'}} \cdot P) = ayvd \cdot P + ayvk_1 sk_{\mathcal{S}} \cdot P + ayvk_1 sk_{\mathcal{C}_{b'}} \cdot P = vd \cdot h_2 + vk_1 sk_{\mathcal{S}} \cdot h_2 + avk_1(x_{\mathcal{C}_{b'}} y \cdot P)$. After that, $\mathcal{S}$ can compute $(avk_1)^{-1}(x_2 - (vd + vk_1 sk_{\mathcal{S}}) \cdot h_2) = x_{\mathcal{C}_{b'}} yP$, which is the solution of the given ECDDHP.

If $\mathcal{S}$ does not abort during the whole simulation process, it needs to not abort all $\mathcal{O}_{\mathcal{S}}$ queries. Firstly, $\mathcal{S}$ queries H_1 for R_2. Due to $d, k_2, u, v \in Z_q^*$, they are uniformly distributed in Z_q^4, H_1's collision probability is $\frac{q_h}{2^{4q}}$. Thus, the probability of fail queries is at most $\frac{q_h(q_{s_1}+q_{s_2})}{2^{4q}}$. And $\mathcal{S}$ queries H_3 for $(R_2, x_2, y_2, pk_{\mathcal{S}}, k_1 v, E, M)$. w is uniformly distributed in Z_q according to $w \in Z_q^*$, thus H_3's collision probability is $\frac{2q_h+q_{s_1}+q_{s_2}}{2^q}$. Finally, we can get that the probability of fail queries is at most $\frac{(q_{s_1}+q_{s_2})(2q_h+q_{s_1}+q_{s_2})}{2^q}$.

In addition, let $event_1$ denote that $\mathcal{A}$ forged a valid signature without querying H_1 queries, $event_2$ denote that $\mathcal{A}$ forged a valid signature with $H_1(v^{-1}R_2) \neq H_1(d \cdot P + k_1 sk_{\mathcal{S}} \cdot P + k_1 sk_{\mathcal{C}_{b'}} \cdot P)$. If $event_1 \cup event_2$ occurs, ECDDHP cannot be solved. The probability of $event_1 \cup event_2$ occurrence is $\Pr[event_1 \cup event_2] = \Pr[event_1 \cap \neg event_2] + \Pr[event_2]$. From $event_1$ and $\neg event_2$, we can get that $[v(d + k_1 sk_{\mathcal{S}} + k_1 x_{b'})]^{-1} x_2 = h_2 = H_1(R_2)$. Due to the selection of all variants in Z_q^*, $\Pr[event_1 \cup event_2]$ is at most $\frac{1}{2^q}$. About $event_2$, let $R_2 = U \cdot P + \sum_{l=1}^{n} pk_l$, $h_2 = H_1(R_2)$, $y_2 = U' \cdot h_2$, $x_2 = t_{\mathcal{C}}' v' \cdot h_2 \neq t_{\mathcal{C}} v \cdot h_2$. Since $sig = \{R_1^{''}, K, k_2, x_2, T\}$ is valid, we can get that $R_2 = u \cdot P - t_{\mathcal{C}} v k_2 \cdot P + \sum_{l=1}^{n} pk_l$, $h_2 = H_1(R_2)$, $y_2 = u \cdot h_2 - k_2 \cdot x_2$, $U = u - k_2 t_{\mathcal{C}} v$, $U' = u - k_2 t_{\mathcal{C}}' v'$. Thus, $k_2 = \frac{U'-U}{t_{\mathcal{C}} v - t_{\mathcal{C}}' v'} = H_3(*)$. Due to the distribution of H_3 queries in Z_q, the probability of querying H_3 for k_2 is at most $\frac{q_h}{2^q}$.

Finally, we can get the probability of $\mathcal{S}$ deciding $x_1 y = x_0 y$ from the forgery is at least $Adv_{\mathcal{A}}^{Anony}(\lambda, q_j, q_p, q_s) = \varepsilon - \frac{q_h(q_{s_1}+q_{s_2})}{2^{4q}} - \frac{(q_{s_1}+q_{s_2})(2q_h+q_{s_1}+q_{s_2})}{2^q} - \frac{1}{2^q} - \frac{q_h}{2^q} - \frac{1}{2} < \varepsilon$.

Based on the above arguments, **Theorem 2** holds. ■

C. PRIVACY ANALYSIS

1) IDENTITY PRIVACY

Firstly, we have proved that the signature scheme adopted in our proposed system satisfies anonymity in Section VI.B **Theorem 2**. That is, when OBU of a vehicle submits a navigation query, RSUs, NSP and other external entities cannot distinguish the OBU's public key pk_{OBU_i} from multiple public keys $\{pk_1, pk_2, \ldots, pk_i, \ldots, pk_n\}$. Secondly, each navigation query of an OBU uses different symmetric key *symk* and public key set. Thus, RSUs, NSP and other external entities cannot analyze and get the public key of the OBU from multiple navigation queries.

Therefore, the proposed system preserves the identity privacy of the vehicles.

2) CURRENT LOCATION PRIVACY

According to GNSS and ground receiver, the OBU calculates and obtains its current location. Firstly, the OBU of a vehicle does not give its current location away to anyone during the navigation query process. Secondly, since the OBU will transmit its navigation query to its nearby RSU, the location of the OBU may be disclosed through the location of the RSU. But the OBU's identity privacy has been preserved. Thus, the RSUs, NSP and external entities cannot know that which OBU is located near the RSU.

Therefore, the proposed system preserves the current location privacy of the vehicles.

3) PRIVACY OF START POINT AND DESTINATION

Firstly, when the OBU transmits the start point and destination to NSP via RSUs, the start point *StaPt* and destination *Dest* are encrypted. RSUs and external entities cannot decrypt to get the start point and destination. Secondly, although NSP can decrypt and get the start point *StaPt* and destination *Dest*, it cannot know which OBU the start point *StaPt* and destination *Dest* information belongs to. Thus, for NSP, a specific OBU's start point and destination privacy is preserved.

Therefore, the proposed system preserves the privacy of start point and destination.

4) ROUTE PRIVACY

Similar to the above privacy of start point and destination, firstly, the route *RT* is also transmitted after encryption. Thus, the RSUs and external entities cannot decrypt and get the route information. Secondly, although NSP generates and knows the optimal route, it cannot know that the route *RT* is sent to which OBU. Thus, for NSP, a specific OBU's route privacy is preserved.

Therefore, the proposed system preserves the route privacy of the vehicle users.

5) UNLINKABILITY

Firstly, in Section VI.B, the anonymity of the signature scheme used in proposed system has been proved. In other words, according to a navigation query, RSUs and external entities cannot link it to a specific OBU. Although NSP can get the start point and destination, it also cannot link them to the OBU. In addition, according to multiple navigation queries and same start points and destinations, RSUs, external entities and NSP also cannot link them to a specific OBU. Because the OBU of a vehicle utilizes different public key set and different symmetric key to sign and encrypt its navigation query respectively.

Therefore, the proposed system realizes the unlinkability.

TABLE 2. Experimental configurations.

Item	Configurations of PC
CPU	12th Gen Intel(R) Core(TM) i5-12400 2.50 GHz
RAM	16.0 GB
OS	Windows 11

VII. PERFORMANCE EVALUATION

In this section, we first measure the basic cryptographic operations and transmission delays. And due to that the systems in [25] and [27] also concerns that the location and route related information cannot be disclosed to NSP except outsiders and inside RSUs, thus we evaluate and compare the computation and transmission overhead in each phase of the proposed system with that of the existing systems [25], [27] to show that our proposed system is efficient and practical.

A. EXPERIMENT SETUP

The proposed system is implemented on personal computer, the configurations of which is presented in TABLE 2 What's more, we use the bilinear pairing (Ate pairing) $e : G \times G \rightarrow G_1$ on a Barreto-Naehrig (BN) curve over a 256-bit prime field. The curve equation is $E(F_p) : y^2 = x^3+3$. The experiments adopt package bn256 [32] and golang elliptic library [33] implemented by using *Go* language. The hash function we adopt in the proposed system is instantiated with SHA-256. We measure each relevant cryptographic operation 1000 times on the experiment platform and take the average value. Transmission delays are set according to [34]. TABLE 3 provides the average execution time of the related cryptographic operations and transmission delays.

B. OVERHEAD ANALYSIS

In this part, we will evaluate and compare the computation overhead of entities in each process of the proposed system with the existing systems [25], [27]. Note that, in the following analysis, n denotes the number of OBUs, m denotes the number of RSUs. Due to the different system model in our proposed system and [25], [27], we use the entities that form navigation services in their system models as NSPs.

TABLE 3. Execute time of cryptographic operations (ms).

Symbols	Operations	Costs
T_{psm}	Point scalar multiplication in G	0.698
T_{pa}	Point addition in G	0.000175
T_h	Hash operation (SHA-256)	0.000508
T_{bp}	Bilinear pairing	3.245
T_{ecdsas}	Signature generation of ECDSA	0.022638
T_{ecdsav}	Signature verification of ECDSA	0.055
T_{enc}	Symmetric encryption (AES)	0.000512
T_{dec}	Symmetric decryption (AES)	0.000511
T_{ex}	Modular exponentiation	0.012
T_{trans}^{or}	Delay between OBU and RSU	≤10
T_{trans}^{rt}	Delay between RSU and TA	≤10
T_{trans}^{ot}	Delay between OBU and TA	≤10
T_{trans}^{nt}	Delay between NSP and TA	≤10
T_{trans}^{rn}	Delay between RSU and NSP	≤10
T_{trans}^{on}	Delay between OBU and NSP	≤20

TABLE 4. Comparison of overhead in system initialization phase (ms).

System	TA	NSP	RSU	OBU
[25]	0.024	-	-	-
[27]	0.024	-	-	-
Ours	0.698	-	-	-

1) SYSTEM INITIALIZATION PHASE

During the *System Initialization* phase, firstly, in our proposed system, only TA executes a point scalar multiplication, i.e. $T_{psm} \approx 0.698$ ms. In both systems of [25] and [27], TA needs to perform twice power operations on Z_q^*, i.e. $2T_{ex} \approx 0.024$ ms.

During the process in this phase of all systems, there is no transmission delay. The overhead comparison of this phase is shown in TABLE 4.

2) REGISTRATION PHASE

During this phase, firstly, in our proposed system, TA needs to perform a point scalar multiplication operation and a transmission operation for registration of NSP and a RSU respectively i.e. $T_{psm} + T_{trans}^{nt}/T_{trans}^{rt} \approx 10.698$ ms. NSP and RSUs need to submit registration request in this process, that is $T_{trans}^{nt}/T_{trans}^{rt} \approx 10$ ms. For an OBU, TA executes four point scalar multiplication operations, two hash operations and a transmission operation for the registration of an OBU, i.e. $4T_{psm} + 2T_h + T_{trans}^{ot} \approx 12.793$ ms. OBU needs to submit a registration request and perform four point scalar multiplication operations, two hash operations and two point addition operations to complete its registration, i.e. $T_{trans}^{ot} + 4T_{psm} + 2T_h + 2T_{pa} \approx 12.793$ ms.

TABLE 5. Comparison of overhead in registration phase (ms).

System	TA	NSP	RSU	OBU
[25]	$10.012+10.012m+13.335n$	10	10	10.059
[27]	$10.012m+26.573n$	-	10	10.103
Ours	$10.698+10.698m+12.793n$	10	10	12.793

In system of [25], TA performs a modular exponentiation operation and a transmission operation for registration of NSP and a RSU respectively, i.e. $T_{ex} + T_{trans}^{nt}/T_{trans}^{rt} \approx 10.012$ ms. NSP and RSUs need to submit registration request in this process, that is $T_{trans}^{nt}/T_{trans}^{rt} \approx 10$ ms. For the registration of an OBU, TA performs a signature verification operation, a bilinear pairing operation, a signature generation operation, a modular exponentiation operation, and a transmission operation, i.e. $T_{ecdsav} + T_{bp} + T_{ecdsas} + T_{ex} + T_{trans}^{ot} \approx 13.335$ ms. OBU in this phase needs to transmit its registration request, and perform three modular exponentiation operations and a signature generation operation, i.e. $T_{trans}^{ot} + 3T_{ex} + T_{ecdsas} \approx 10.059$ ms.

In system of [27], there is no NSP role. For the registration of a RSU, TA performs a modular exponentiation operation and a transmission operation, i.e. $T_{ex} + T_{trans}^{rt} \approx 10.012$ ms. RSUs need to submit registration request in this process, that is $T_{trans}^{rt} \approx 10$ ms. TA needs to perform five modular exponentiation operations, two bilinear pairing operations, a signature generation operation and two transmission operations for an OBU's registration, i.e. $5T_{ex} + 2T_{bp} + T_{ecdsas} + 2T_{trans}^{ot} \approx 26.573$ ms. During the process, OBU performs four modular exponentiation operations, a signature verification operation, and an operation of transmission to TA, i.e. $4T_{ex} + T_{ecdsav} + T_{trans}^{ot} \approx 10.103$ ms.

The overhead comparison of this phase is shown in TABLE 5 and the results have been rounded to three decimal places.

3) NAVIGATION SERVICE PROVISION PHASE

Firstly, during the navigation service provision phase in our proposed system, OBU needs to execute seven point scalar multiplication operations, three hash operations, a symmetric encryption operation and $n + 1$ point addition operations, a transmission to RSU operation and a symmetric decryption operation, i.e. $7T_{psm} + 3T_h + T_{enc} + (n+1)T_{pa} + T_{trans}^{or} + T_{dec} \approx 14.889$ ms. In this phase, RSU performs four point scalar multiplication operations, $n + 2$ point addition operations, two hash operations, a signature generation operation, a transmission to NSP operation and a transmission to OBU operation, i.e. $4T_{psm}+(n+2)T_{pa}+2T_h+T_{ecdsas}+T_{trans}^{rn}+T_{trans}^{or} \approx 22.816+0.000175n$ ms. NSP in this phase performs a signature verification operation, a point scalar multiplication operation, a hash operation, a symmetric decryption and a symmetric encryption operation. In addition, it needs to transmit information to RSU once. That is $T_{ecdsav} + T_{psm} + T_h + T_{dec} + T_{enc} + T_{trans}^{rn} \approx 10.755$ ms.

TABLE 6. Comparison of overhead in navigation service provision phase (ms).

System	TA	NSP	RSU	OBU
[25]	-	$0.049+0.001n+0.055m$	$39.793+9.736n$	13.350
[27]	-	42.828	$30.024+9.736n$	23.041
Ours	-	10.755	22.816	14.889

In system of [25], OBU needs to perform a symmetric encryption operation, two hash operations, four modular exponentiation operations, a bilinear pairing operation, a transmission to RSU operation and a signature verification operation, i.e. $T_{enc} + 2T_h + 4T_{ex} + T_{bp} + T_{trans}^{or} + T_{ecdsav} \approx 13.350$ ms. RSU needs to perform $3n + 3$ bilinear pairing operations, $n + 1$ hash operations, a modular exponentiation operation, two signature generation operations, a transmission to crowd-sourcing OBUs operation, a transmission to NSP operation, and a transmission to OBU operation i.e. $(3n + 3)T_{bp} + (n + 1)T_h + T_{ex} + 2T_{ecdsas} + 2T_{trans}^{or} + T_{trans}^{rn} \approx (39.793 + 9.736n)$ ms. For NSP, it executes $n+3$ hash operations, two modular exponentiation operations, a symmetric encryption operation, m signature verification operations, a signature operation and a transmission to RSU operation i.e. $(n+3)T_h + 2T_{ex} + T_{enc} + mT_{ecdsav} + T_{ecdsas} + T_{trans}^{rn} \approx (0.049+0.001n + 0.055m)$ ms.

In system of [27], OBU costs five modular exponentiation operations, two hash operations, four bilinear pairing operations and a transmission operation, i.e. $5T_{ex} + 2T_h + 4T_{bp} + T_{trans}^{or} \approx 23.041$ ms. RSU costs $3n$ bilinear pairing operations, n hash operations, two modular exponentiation operations, a transmission to crowd-sourcing OBUs operation, two transmission to NSP operations, i.e. $3nT_{bp} + nT_h + 2T_{ex} + T_{trans}^{or} + 2T_{trans}^{rn} \approx (30.024+9.736n)$ ms. NSP costs seven bilinear pairing operations, four hash operations, nine modular exponentiation operations, four symmetric encryption operations, one symmetric decryption operation and two transmission to RSU operations, i.e. $7T_{bp} + 4T_h + 9T_{ex} + 4T_{enc} + T_{dec} + 2T_{trans}^{or} \approx 42.828$ ms.

The overhead comparison of this phase is shown in TABLE 6, and the results have been rounded to three decimal places.

From TABLE 4 we can see that the computation overhead of TA in our proposed system is larger than that in [25] and [27]. But System Initialization phase is only performed once by TA during the whole process of system running.

From TABLE 5 we can get that the overhead of TA is related with the value of m and n. And the overhead of OBU in our proposed system is also larger than that in [25] and [27]. Similarly, the *Registration* phase is also performed once during the whole process of system running. In addition, this phase and *System Initialization* phase are performed before providing navigation service, and they can be executed offline. Therefore, the two phases mentioned above will not affect the efficiency of navigation service provision.

From TABLE 6 we can see that the overhead of NSP in [25] is related with the value of m and n. And the overhead of NSP in our proposed system is significantly less than that in [27]. The overhead of RSU in our proposed system is significantly less than that in [25] and [27]. The overhead of OBU in our proposed system and [25] is significantly less than that in [27]. And the overhead of OBU in our proposed system slightly larger than that in [25]. But the overhead of whole *Navigation Service Provision* phase in our proposed system is 48.460 ms, which is obviously less than that in [25] and [27]. They are $(53.192+9.737n + 0.055m)$ ms and $(95.893+9.736n)$ ms respectively.

Therefore, the efficiency of our proposed system is more efficient comparing with the systems in [25] and [27].

VIII. CONCLUSION AND FUTURE WORK

In this paper, we propose a dual privacy-preserving lightweight navigation system for vehicular networks, which achieves the unlinkable anonymity of vehicles for outsiders and inside RSUs and NSP and unlinkability between the location, start point, destination, route information and PII to preserve the privacy of the route related information. We conduct the correctness, security and privacy analysis of the proposed system. In addition, the performance is evaluated through conducting the experiments on the Desktop computer. The results show that our proposed system is more efficient for providing navigation service comparing with existing systems.

In the future, we are going to explore whether our system can combine lightweight zero knowledge proof technology to realize more personal information hiding and efficiency improvement.

REFERENCES

[1] *World Urbanization Prospects: The 2014 Revision, Highlights. Department of Economic and Social Affairs*, Population Division, United Nations, New York, NY, USA, 2014.

[2] Y.-T. Tseng and H.-W. Ferng, "An improved traffic rerouting strategy using real-time traffic information and decisive weights," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 9741–9751, Oct. 2021.

[3] N. Kaloudis, "Security and privacy issues in smart city infrastructure with emphasis on mobility," M.S. thesis, Univ. Aegean, 2018.

[4] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. Shen, "Privacy-preserving smart parking navigation supporting efficient driving guidance retrieval," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6504–6517, Jul. 2018.

[5] L. Zhu, M. Li, Z. Zhang, and Z. Qin, "ASAP: An anonymous smart-parking and payment scheme in vehicular networks," *IEEE Trans. Depend. Secure Comput.*, vol. 17, no. 4, pp. 703–715, Jul. 2020.

[6] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 772–782, Jun. 2017.

[7] M. Li, L. Zhu, and X. Lin, "Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4573–4584, Jun. 2019.

[8] M. Nabil, A. Sherif, M. Mahmoud, A. Alsharif, and M. Abdallah, "Efficient and privacy-preserving ridesharing organization for transferable and non-transferable services," *IEEE Trans. Depend. Secure Comput.*, vol. 18, no. 3, pp. 1291–1306, May 2021.

[9] Y. Luo, X. Jia, S. Fu, and M. Xu, "PRide: Privacy-preserving ride matching over road networks for online ride-hailing service," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1791–1802, Jul. 2019.

[10] J. Ni, X. Lin, K. Zhang, and X. Shen, "Privacy-preserving real-time navigation system using vehicular crowdsourcing," in *Proc. IEEE 84th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2016, pp. 1–5.
[11] M. Li, L. Zhu, and X. Lin, "Privacy-preserving traffic monitoring with false report filtering via fog-assisted vehicular crowdsensing," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 1902–1913, Nov. 2021.
[12] *Google Maps*. Accessed: Sep. 2, 2022. [Online]. Available: http://www.google.cn/maps
[13] Y. Zhang, C. Xu, X. Lin, and X. Shen, "Blockchain-based public integrity verification for cloud storage against procrastinating auditors," *IEEE Trans. Cloud Comput.*, vol. 9, no. 3, pp. 923–937, Jul. 2021.
[14] Y. Yao, X. Chang, J. Misic, V. B. Misic, and L. Li, "BLA: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3775–3784, Apr. 2019.
[15] J. Camenisch, M. Drijvers, A. Lehmann, G. Neven, and P. Towa, "Zone encryption with anonymous authentication for V2V communication," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Sep. 2020, pp. 405–424.
[16] S. Lv and Y. Liu, "PLVA: Privacy-preserving and lightweight V2I authentication protocol," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6633–6639, Jul. 2022.
[17] Y. Yang, L. Wei, J. Wu, C. Long, and B. Li, "A blockchain-based multidomain authentication scheme for conditional privacy preserving in vehicular ad-hoc network," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8078–8090, Jun. 2022.
[18] X. Yue, S. Zeng, X. Wang, L. Yang, S. Bai, and Y. He, "A practical privacy-preserving communication scheme for CAMs in C-ITS," *J. Inf. Secur. Appl.*, vol. 65, Mar. 2022, Art. no. 103103.
[19] B. Chaudhary and K. Singh, "A blockchain enabled location-privacy preserving scheme for vehicular ad-hoc networks," *Peer Peer Netw. Appl.*, vol. 14, no. 5, pp. 3198–3212, Sep. 2021.
[20] Y. Wang, X. Li, X. Zhang, X. Liu, and J. Weng, "ARPLR: An all-round and highly privacy-preserving location-based routing scheme for VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16558–16575, Sep. 2022.
[21] V. K. Yadav, S. Verma, and S. Venkatesan, "Linkable privacy-preserving scheme for location-based services," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7998–8012, Jul. 2022.
[22] Y. Yao, X. Chang, J. Wang, J. Mišić, V. B. Mišić, and H. Wang, "LPC: A lightweight pseudonym changing scheme with robust forward and backward secrecy for V2X," *Ad Hoc Netw.*, vol. 123, Dec. 2021, Art. no. 102695.
[23] P. Zhang, C. Hu, D. Chen, H. Li, and Q. Li, "ShiftRoute: Achieving location privacy for map services on smartphones," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4527–4538, May 2018.
[24] J. Ni, X. Lin, and X. Shen, "Toward privacy-preserving valet parking in autonomous driving era," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2893–2905, Mar. 2019.
[25] M. Li, Y. Chen, S. Zheng, D. Hu, C. Lal, and M. Conti, "Privacy-preserving navigation supporting similar queries in vehicular networks," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 2, pp. 1133–1148, Mar./Apr. 2022.
[26] H. Zhong, J. Ni, J. Cui, J. Zhang, and L. Liu, "Personalized location privacy protection based on vehicle movement regularity in vehicular networks," *IEEE Syst. J.*, vol. 16, no. 1, pp. 755–766, Mar. 2022.
[27] B. Baruah and S. Dhal, "A security and privacy preserved intelligent vehicle navigation system," *IEEE Trans. Depend. Secure Comput.*, early access, Jan. 25, 2022, doi: 10.1109/TDSC.2022.3145649.
[28] D. Hankerson, J. A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Cham, Switzerland: Springer, 2006.
[29] H. Krawczyk, "Cryptographic extraction and key derivation: The HKDF scheme," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2010, pp. 631–648.
[30] O. Goldreich, *Foundations of Cryptography: Basic Applications*, vol. 2. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[31] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.
[32] *Package Bn256*. Accessed: Sep. 13, 2022. [Online]. Available: https://godoc.org/github.com/cloudare/bn256
[33] *Golang Elliptic Library*. Accessed: Sep. 13, 2022. [Online]. Available: https://github.com/glycerine/fast-elliptic-curve-p256.git
[34] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2627–2637, Aug. 2018.

YINGYING YAO received the Ph.D. degree in cyberspace security computer science and technology from Beijing Jiaotong University, in 2021. She is currently a Postdoctoral Researcher at the School of Computer and Information Technology, Beijing Jiaotong University. Her current research interests include network security and privacy, applied cryptography, and blockchain.

XIAOLIN CHANG (Senior Member, IEEE) received the Ph.D. degree in computer science from The Hong Kong University of Science and Technology, in 2005. She is currently a Professor at the School of Computer and Information Technology, Beijing Jiaotong University. Her current research interests include edge/cloud computing, network security, security, and privacy in machine learning.

LIN LI received the Ph.D. degree from Shandong University, in 2007. She is currently an Associate Professor at the Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University. Her current research interests include cryptography and federated learning.

JIQIANG LIU (Senior Member, IEEE) received the Ph.D. degree from Beijing Normal University, in 1999. He is currently a Full Professor and the Dean of the School of Software Engineering, Beijing Jiaotong University. He has authored or coauthored over 100 publications. In recent years, he has been mainly engaged in research on trusted computing, privacy protection, and cloud computing security.

HONG WANG is currently an Associate Professor with the Henan Key Laboratory of Network Cryptography Technology. His research interests include information security and quantum cryptography.