Cloud-Based Privacy-Preserving Parking Navigation Through Vehicular Communications

Jianbing Ni[1], Kuan Zhang[1], Xiaodong Lin[2], Yong Yu[3](✉), and Xuemin (Sherman) Shen[1]

[1] Department of Electrical and Computer Engineering,
University of Waterloo, Waterloo, ON N2L 3G1, Canada
{j25ni,k52zhang,sshen}@uwaterloo.ca

[2] Faculty of Business and Information Technology,
University of Ontario Institute of Technology, Oshawa, ON L1H 7K4, Canada
xiaodong.lin@uoit.ca

[3] School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu 611731, China
yuyong@uestc.edu.cn

Abstract. Finding a vacant parking space in a congested area, such as shopping mall, airport, etc., is always time-consuming and frustrating for drivers. Real-time parking information can avoid vehicles being cruising on the roads. However, when the drivers are acquiring park ing information, their privacy is inevitable to be disclosed. In this paper, to minimize drivers' hassle and preserve drivers' privacy, we propose CPARN, a Cloud-based Privacy-preserving pARking Navigation system through vehicular communications, in which a cloud server guides drivers to vacant parking spaces close to their desired destinations without exposing the privacy of drivers, including drivers' identities, references and routes. Specifically, CPARN allows drivers to query vacant parking spaces in an anonymous manner to a cloud server that maintains the parking information, and retrieve the protected navigation responses from the roadside units when the vehicles are passing through. CPARN has the advantage that it is unnecessary for a vehicle to keep connected with the queried roadside unit to ensure the retrievability of the navigation result, such that the navigation retrieving probability can be significantly improved. Performance evaluation through extensive simulations demonstrates the efficiency and practicality of CPARN.

Keywords: Vehicular ad hoc networks (VANETs) · Parking navigation · Privacy preservation · Vehicular communications

1 Introduction

With the large number of vehicles in metropolises, parking in a congested area, e.g., downtown, shopping mall, particularly in peak hours, has become

Y. Yu—This work is supported by the Fundamental Research Funds for the Central Universities under Grant ZYGX2015J059, ZYGX2014J062.

R. Deng et al. (Eds.): SecureComm 2016, LNICST 198, pp. 85–103, 2017.
DOI: 10.1007/978-3-319-59608-2_5

a conflicting and confusing issue for a number of drivers [1]. It is common for vehicles to cruise among parking lots or circle within a large parking lot for an accessible parking spot. In crowded area, such vehicles cause an average 30% of the traffic on the road [2]. This situation becomes worse in some developing countries, such as China and India, where the number of the parking facilities is not sufficient for private vehicles. The extra traffic leads to significant social problems, such as traffic congestions, fuel waste, air pollution and vehicle accidents. Although traditional navigation systems, e.g., Google map and on-board navigation systems, can assist to locate parking garages, drivers may still worry about whether there is available parking space when they arrive, specifically in peak hours and congested area. Parking guidance information systems [3,4] can broadcast the availability of parking spaces at some specific spots or on the Internet. However, the former method may increase the traffic pressure around these spots, and the latter approach is unrecommended since it is dangerous for the drivers to use mobile devices when driving.

Recently, vehicular ad hoc networks (VANETs) become increasingly popular in both industry and academia [5,6]. In VANET, each vehicle, equipped with an onboard unit (OBU) device, is allowed to communicate with other vehicles nearby, e.g., vehicle-to-vehicle (V2V) communications, and with roadside units (RSUs), i.e., vehicle-to-roadside (V2R) communications [7,8]. VANET-based parking navigation systems can provide real-time parking navigation services for drivers on roads. By means of the widely deployed vehicular communication infrastructure, the vehicles can use OBUs to acquire the real-time parking information. Specifically, a vehicle can query the available parking space near the destination through the nearby RSUs and obtain the up-to-date information to find the accessible parking lot. Such a parking navigation system has the advantage that the driver can conveniently enjoy real-time parking navigation services and reach available parking space within short time and low fuel cost.

However, security and privacy issues are preliminary concerns for drivers in VANETs, since the infrastructure is confronted with various malicious attacks. If these issues cannot be well addressed, it is impossible for drivers to adopt the parking navigation services. To prevent attackers from submitting invalid queries to the RSUs, registration is necessary for the navigation services. The drivers must be authenticated to make sure that they are the registers, such that it is feasible for detecting a fabricated vehicle, which pretends a legal vehicle to enjoy free services. Besides, to ensure the trustworthiness of the interactions, all messages sent by vehicles and RSUs must be signed to guarantee that they are not polluted or forged by the attackers. Confidentiality of queries and responses is another security issue for VANET-based parking navigation systems. A driver does not want other drivers nearby to learn the destination by eavesdropping on the queries. Furthermore, the navigation response should not be shared with all vehicles nearby if this service is charged; otherwise, the vehicles can enjoy free parking navigation services, in case they have the same destination with the querying vehicle.

Location privacy is another concern for drivers, and there have been numerous controversies due to the track exposure [9,10]. For example, some navigation applications, offered by Google and Apple, collect drivers' locations and destinations [11], which may reveal sensitive information about the drivers' personal lives. In VANET-based parking navigation systems, the OBUs on vehicles frequently communicate with RSUs to query and receive the parking information. The vehicle's location is inevitable to be exposed, which is tightly related to the driver however. An attacker can learn the routes of vehicles and predict the location of drivers at a specific time, and even identify the references, health condition, social and political affiliations based on the visiting frequency of specific places. Moreover, the disclosure of vehicles' location may bring significant convenience for car thieves, since the thieves may trace the vehicles several days before action and prefer to steal cars parking in quiet places [12]. Therefore, location privacy is a must to be preserved for the wide acceptance of navigation service to the public. One common approach of location privacy leakage-resilient is to keep the drivers anonymous using pseudonyms or anonymous credentials. As a result, no attacker can identify the identities of drivers or link navigation messages to reconstruct the route of a specific driver. Nevertheless, once the drivers' identities are preserved, it is impossible to return the navigation responses to the target vehicles. To address the contradiction between identity privacy preservation and navigation responses retrievability, Chim et al. [13] require the vehicle to keep the connection alive with the RSU after sending the navigation query until it successfully obtains the reply, which is quite challenging in reality, particularly, when the vehicle moves at a pretty high speed. As a result, the successful delivery probability of navigation responses is limited. In addition, full anonymity is not perfect because a vehicle may launch a denial-of-service attack by sending a large number of queries to the RSUs in a short period of time. The misbehaving drivers should be traced when necessary.

In this paper, we propose a Cloud-based Privacy-preserving pARking Navigation (CPARN) system by integrating vehicular communications and a cloud server, which provides navigation service to assist drivers to find available parking spots efficiently. In specific, a driver can query a vacant parking spot by submitting his/her current location and desired destination to the cloud server. Then, the server automatically searches for an available parking lot close to the destination and vacant parking spots in the recommended lot using the real-time parking information outsourced by parking lots. Finally, the server returns the navigation response to the driver through RSUs on the way to his/her destination. This is reasonable because most of drivers use GPS, so that the driving direction to a destination from an area can be predicted. As a result, the RSUs that the driver will pass through can also be determined, and thereby receive the navigation response successfully. The contributions of this paper are four-fold:

- We propose CPARN based on VANETs to achieve parking navigation for drivers. With the parking navigation offered by the cloud server, a vehicle can quickly find a vacant parking space close to the desired destination. The gasoline and the time wasted on searching for parking spaces can be reduced.
- CPARN achieves conditional privacy preservation for drivers by utilizing anonymous credentials. Specifically, an authenticated vehicle sends the

parking navigation query to the cloud server without exposing the real identity. Meanwhile, a trusted authority can trace the identity of a misbehaving vehicle.

- We propose a novel approach to improve navigation retrieving probability in anonymous vehicular communications. We do not require the vehicle to communicate with the same RSU in the query and response procedures. Instead, the driver can send parking navigation query to the cloud server through a nearby RSU, and search and retrieve the response from the RSUs built on the driving routes. In this case, the communication delay is tolerable, and the probability that vehicles can retrieve the navigation responses successfully can be dramatically improved. Note that the new method is still suitable for the situation where the response of the query is returned rapidly, and thus, the vehicle can retrieve the navigation response from the queried RSU.
- We discuss the security features and evaluate the performance of CPARN. The extensive simulations demonstrate that the system is efficient and practical.

The remainder of this paper is organized as follows. In Sect. 2, we formalize system model, threat model and security goals. In Sect. 3, we propose the CPARN system, followed by security discussion in Sect. 4, and the performance evaluation in Sect. 5, respectively. We review the related work in Sect. 6 and conclude our paper in Sect. 7.

2 Problem Statement

In this section, we state the problem by formalizing the system and threat models, and identify the security goals.

2.1 System Model

We consider the system model of the parking navigation service, which consists of a trusted authority (TA), a cloud, parking lots, a large number of vehicles and some RSUs.

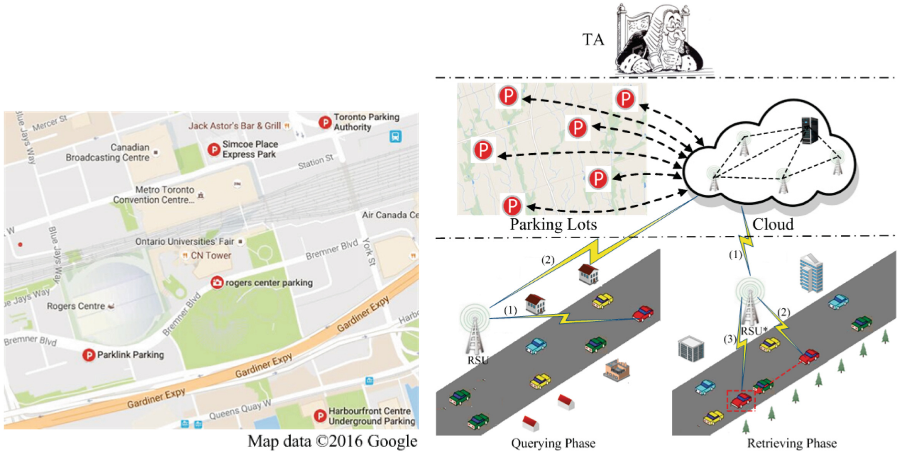
- *TA.* The TA is a trusted party, whose responsibility is to generate the public key certificates for all the entities in the system, and to trace the identities of vehicles when necessary.
- *Cloud.* The cloud, which consists of a server and connected RSUs, can provide two types of services. One service is offering the real-time parking data storage for the parking lots in a specific area; the other is providing the parking navigation for drivers by using the maintaining real-time parking data. For example, the red points in Fig. 1(a) are parking lots around CN tower, and the cloud stores the parking data for these lots and navigates for the drivers whose destination is CN tower.
- *Parking Lots.* Parking lots offer parking spots to vehicles. To manage the parking spaces and charge the parking fee, parking lots record the real-time occupancy of each parking space, and outsource their data to the cloud to reduce the cost of data management and maintenance.

- *Vehicles.* Each vehicle is equipped with an irreplaceable and temper-proof OBU, which provides the capacity to communicate with the nearby vehicles and the RSUs. OBUs can also execute some simple computations and have a small amount of read-only memory.
- *RSUs.* RSUs are deployed on the road, which can communicate with each other and with vehicles driving through. They can also interact with the cloud and the TA via the Internet. Each RSU is resource in rich, indicating that it has enough storage space to maintain the navigation responses and computational capacity to perform the cryptographic operations.

Figure 1(b) shows the system model of parking navigation service. Firstly, the cloud server, vehicles and RSUs generate the public-secret key pairs and register the public key certificates at TA, respectively. The cloud offers parking data storage service to parking lots and the parking lots outsource their real-time parking data to the cloud through the Internet. As there is no security and privacy issues for the parking lots, the data storage service is beyond our work. To make fully use of the real-time parking data, the cloud provides parking navigation for drivers. To participate the parking navigation service, each vehicle needs to register at the cloud server and obtain an anonymous credential to access the service. The parking navigation consists of two phases: querying and retrieving. In the querying phase, a vehicle firstly generates and sends a parking query to the nearby RSU (Step 1). Upon receiving a query, the RSU forwards it to the cloud server (Step 2). The cloud server recommends an accessible parking lot to the vehicle according to the real-time parking information and the desired destination of the vehicle. In the retrieving phase, the cloud server firstly sends the navigation response to the RSUs located on the roads that the querying vehicle may drive through (Step 1). The RSUs store the navigation responses on a navigation table temporarily after receiving the messages from the cloud server. When the vehicle enters the coverage area of an RSU, it sends a retrieving query to the RSU (Step 2). Upon receiving the retrieving query from a vehicle, the RSU searches the navigation response and returns it to the vehicle if the RSU is maintaining the response; otherwise, the RSU returns failure and the vehicle tries to retrieve the response from the next RSU (Step 3).

2.2 Threat Models

The threats may be from internal and external attackers. The external attackers may compromise the cloud server and RSUs to steal sensitive information about drivers. The eavesdroppers can listen on the communication channels and capture the transmitting messages to analyze driver's references. Internal threats come from the curious employees in cloud or drivers who want to learn more information about other drivers. Therefore, the whole infrastructure is confronted with a variety of security threats and no entity can be fully-trusted except the TA. Although the cloud server has to follow the regulations and agreements that are agreed with the vehicles, it is also interested in drivers' privacy and eager to mine private knowledge from the parking navigation queries.



Map data ©2016 Google

(a) Parking around CN Tower, Toronto (b) System Model

Fig. 1. System model of CPARN

The vehicles may be compromised and launch some attacks to the cloud server, e.g., denial-of-service attack, impersonation attack and replay attack. Besides, they are also curious about the driving routes of the nearby drivers. The RSUs may also be compromised and the attackers can obtain the navigation messages maintained on storage devices. The RSUs are interested in the drivers' privacy and try to learn information by analyzing the forwarding data, e.g., navigation queries and navigation responses.

2.3 Security Goals

We aim to construct a system, which can provide real-time parking navigation, to achieve the following security goals:

- Service Authentication. A vehicle should be authenticated before submitting the parking navigation query, such that no attacker can impersonate a registered vehicle to enjoy free navigation service if the service is charged.
- Message Authentication and Integrity. The cloud server, the RSUs and the vehicles should ensure that the sent messages, including the navigation queries and responses, would not be polluted or forged by attackers. Thus, the receivers can believe the genuine of the messages.
- Identity Privacy Preservation. The identities of the drivers should be well-protected against the cloud server, the RSUs and other vehicles during the parking navigation procedure. Moreover, given two navigation queries, neither the cloud server nor the RSUs can identify whether these queries are sent by the same vehicle.
- Confidentiality. The contents of a navigation query and the corresponding response should be confidential to the vehicles nearby, the RSUs and the

eavesdroppers. Even the compromised RSUs cannot learn any knowledge about the navigation queries and responses.

- Traceability. The TA can trace the real identities of the vehicles who submit the parking navigation queries to the cloud server. Furthermore, to prevent the denial-of-service attack, the identity of the vehicle who submits more than two different parking navigation queries in a time period, e.g., one second, can be recovered by the cloud server.

3 The CPARN System

In this section, we demonstrate the preliminaries and describe CPARN in detail.

3.1 Preliminaries

If S is a non-empty set, $s \in_R S$ denotes s is randomly chosen from S. $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ is a set of cyclic groups of the same prime order p. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is type 3 bilinear pairing [14], in which $\mathbb{G}_1 \neq \mathbb{G}_2$ and there is no efficiently computable homomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ in either direction.

PS Signature. The PS signature is proposed by Pointcheval and Sanders [14], which has the same features as the CL signature [15], but is more efficient than the CL signature due to the advantage of using type 3 pairing. The existential unforgeability of the PS signature under chosen message attacks can be reduced to the modified LRSW Assumption 2 [14].

Let $\hat{g}$ be a generator of $\mathbb{G}_2$. The secret key of the signer is $(x, y_1, \cdots, y_r) \in_R \mathbb{Z}_p^{r+1}$ and the public key is $(\widehat{X}, \widehat{Y}_1, \cdots, \widehat{Y}_r) \leftarrow (\hat{g}^x, \hat{g}^{y_1}, \cdots, \hat{g}^{y_r})$. A signature on multi-block messages $(m_1, \cdots, m_r) \in \mathbb{Z}_p^r$ is $\sigma = (\sigma_1, \sigma_2) = (h, h^{x+\sum_{j=1}^{r} y_j m_j})$, where h is randomly chosen from $\mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$. The signature σ can be verified publicly as $\sigma_1 \neq \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and $\hat{e}(\sigma_1, \widehat{X} \prod_{j=1}^{r} \widehat{Y}_j^{m_j}) = \hat{e}(\sigma_2, \hat{g})$.

3.2 The CPARN System

Our proposed CPARN consists of five phases: system setup, vehicle registration, navigation querying, response retrieving and vehicle tracing. The details of the CPARN are described as follows.

System Setup. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be three cyclic groups of the same large prime order p. Suppose that $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are equipped with type 3 pairing, that is, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. g is a generator of the group $\mathbb{G}_1$ with $g \neq 1_{\mathbb{G}_1}$, and $\hat{g}, \hat{g}_0$ are two generators of the group $\mathbb{G}_2$ with $\hat{g} \neq \hat{g}_0 \neq 1_{\mathbb{G}_2}$. Define a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_p$. $\mathcal{C} = AES_{ENC}(\mathcal{K}, \mathcal{M})$ and $\mathcal{M} = AES_{DES}(\mathcal{K}, \mathcal{C})$ denote the encryption and decryption algorithms of AES scheme, respectively. The TA chooses $(x, x_1) \in_R \mathbb{Z}_p^2$ and computes $\widehat{X} = \hat{g}^x$, $\widehat{X}_1 = \hat{g}^{x_1}$. The secret key of the TA is (x, x_1) and the public key is $(g, \hat{g}, \widehat{X}, \widehat{X}_1)$.

The cloud server randomly chooses $(y, y_1, y_2, y_3) \in_R \mathbb{Z}_p^4$ and computes

$$(Y, Y_1, Y_2, Y_3, \widehat{Y}, \widehat{Y}_1, \widehat{Y}_2, \widehat{Y}_3) \leftarrow (g^y, g^{y_1}, g^{y_2}, g^{y_3}, \hat{g}^y, \hat{g}^{y_1}, \hat{g}^{y_2}, \hat{g}^{y_3}).$$

The secret key of the cloud server is (y, y_1, y_2, y_3, Y), and the public key is $(Y_1, Y_2, Y_3, \widehat{Y}, \widehat{Y}_1, \widehat{Y}_2, \widehat{Y}_3)$.

Each RSU has a unique number RID associated with its location. The RSU chooses a random number $z \in_R \mathbb{Z}_p$ as its secret key and computes $Z = g^z$ as its public key. In addition, the RSU defines three Bloom filters. BF_K is a (m, n, k, H)-Bloom filter, CBF_K is a (m, n, k, H, λ)-counting Bloom filter and VBF_K is a variant of the Bloom filter. In these Bloom filters, k hash functions $h_l \in H$ are defined as $h_l : \mathbb{G}_1 \rightarrow \mathbb{Z}_m$, for $1 \leq l \leq k$. The difference between VBF_K and the traditional Bloom filter is that instead of using an array of bits to represent the set membership in Bloom filter, VBF_K uses an array of γ-bit strings to indicate the storage addresses of the navigation messages. Every storage address S is divided into k shares of γ-bit, $S_1, S_2, \cdots, S_k$, using the XOR-based secret sharing scheme, and each share is stored on one index in VBF_K according to the hash values of the input. Initially, the array in BF_K, the counters in CBF_K and the strings in VBF_K are set to be zero.

Each vehicle has a unique identity VID. To register on the TA, a vehicle chooses two random $(v, v') \in_R \mathbb{Z}_p^2$ and computes

$$(V, \widehat{V}, \widehat{V}', \widehat{V}_0) \leftarrow (g^v, \widehat{X}_1^v \widehat{X}^{v'}, \hat{g}^{v'}, \widehat{X}_1^v).$$

It sends $(VID, V, \widehat{V}, \widehat{V}')$ to the TA, along with the zero-knowledge proof:

$$\mathcal{PK}_1 = \{(v, v') : V = g^v \wedge \widehat{V} = \widehat{X}_1^v \widehat{X}^{v'} \wedge \widehat{V}' = \hat{g}^{v'}\}.$$

The TA firstly computes $\widehat{V}_1 = \widehat{V} / \widehat{V}'^x$. Then, the TA verifies the validity of the proof $\mathcal{PK}_1$ and checks the equation $\hat{e}(V, \widehat{X}_1) = \hat{e}(g, \widehat{V}_1)$. If either is invalid, the TA returns failure and aborts. Otherwise, the TA generates a random $w \in_R \mathbb{Z}_p$ to calculate

$$(B_1, B_2, B_3) \leftarrow (g^w, (g^x V^{x_1})^w, \hat{e}(B_1, \widehat{X}_1)).$$

(B_1, B_2) is a valid PS signature on v and B_3 is a pre-computed item that allows the vehicle to avoid the bilinear pairing computation during the signing procedure. Finally, the TA returns (VID, B_1, B_2, B_3) to the vehicle through secure channel and stores $(VID, V, \widehat{V}_1)$ in a secret database. Upon getting the response, the vehicle sets its secret key as $(v, \widehat{V}_0, B_1, B_2)$ and the corresponding public key as (V, B_3). The key pair is stored in the read-only memory of the OBU.

Vehicle Registration. To enjoy the parking navigation service, a vehicle should register on the cloud server to obtain an anonymous credential. The vehicle with an identity VID selects two random $(t, s) \in_R \mathbb{Z}_p^2$ to compute $C = g^t Y_1^{VID} Y_2^s Y_3^v$, and sends (VID, C, V) to the cloud server, along with the zero-knowledge proof:

$$\mathcal{PK}_2 = \{(t, s, v) : C = g^t Y_1^{VID} Y_2^s Y_3^v \wedge V = g^v\}.$$

When the cloud server receives the message, it checks the validity of $\mathcal{PK}_2$. If it is invalid, the cloud server returns failure and aborts; otherwise, it chooses a random $u \in_R \mathbb{Z}_p$ to compute

$$(A_1, A_2) \leftarrow (g^u, (YC)^u).$$

Then, the cloud server returns (A_1, A_2) to the vehicle through secure channel and stores (VID, C, V, A_1, A_2) in its database. The vehicle checks

$$\hat{e}(A_1, \widehat{Y})\hat{e}(A_1, \hat{g}^t \widehat{Y}_1^{VID} \widehat{Y}_2^s \widehat{Y}_3^v) \stackrel{?}{=} \hat{e}(A_2, \hat{g}).$$

If yes, the vehicle calculates $A_3 = A_2/A_1^t$, and obtains the anonymous credential $AC = (A_1, A_3)$. Finally, it stores (AC, s) in the read-only memory of the OBU.

Navigation Querying. When a vehicle with the identity VID and the anonymous credential AC is on the road, the driver submits a parking navigation query to the cloud server to find a vacant parking space close to the desired destination. The vehicle generates the basic query information, including the destination $DEST$, current location CL, acceptable price range AP, current time t_1, expected arrival time t_2, expiration time t_3, etc., and performs the following steps to generate a parking navigation query:

- Pick a random $\kappa \in_R \mathbb{Z}_p$ to compute a temporary session key $U = \hat{g}^{\kappa}$ and calculate $L = \mathcal{H}(DEST, CL, AP, t_1, t_2, t_3, N)$ and $T = \hat{g}^{vt_1}\hat{g}_0^{Ls}$, where N is a random number chosen from $\mathbb{Z}_p$.
- Choose two random $(\alpha, \beta) \in_R \mathbb{Z}_p^2$ to compute $AC' = (A'_1, A'_3) = (A_1^{\alpha}, A_3 A_1^{\beta})^{\alpha}$ and generate a zero-knowledge proof as

$$\mathcal{SPK}\left\{\begin{array}{c} (VID, v, s, \kappa, \beta) : \hat{e}(A'_3, \hat{g}) = \hat{e}(A'_1, \widehat{Y})\hat{e}(A'_1, \widehat{Y}_1^{VID}\widehat{Y}_2^s\widehat{Y}_3^v)\hat{\ }(A'_1, \hat{g})^{\beta} \\ \wedge U = g^{\kappa} \\ \wedge\, T = \hat{g}^{vt_1}\hat{g}_0^{Ls} \end{array}\right\}(N).$$

- Encrypt $(DEST, CL, AP, t_2, t_3)$ by selecting two random $r \in_R \mathbb{Z}_p$ and $r_1 \in \mathbb{G}_1$, and computing $c_1 = g^r$, $c_2 = r_1 Y_1^r$, and $c_3 = AES_{ENC}(r_1, DEST|| CL||AP||t_2||t_3)$.
- Randomise (B_1, B_2, B_3) by selecting two random $(r', r'') \in_R \mathbb{Z}_p^2$ and computing

$$(\widetilde{B}_1, \widetilde{B}_2, \widetilde{B}_3) \leftarrow (B_1^{r'}, B_2^{r'}, B_3^{r'r''}),$$

  calculate $c = \mathcal{H}(\widetilde{B}_1, \widetilde{B}_2, \widetilde{B}_3, N, t_1, U, T, AC', \mathcal{SPK}, c_1, c_2, c_3)$, $\tau = r'' + cv$, and output $(\widetilde{B}_1, \widetilde{B}_2, c, \tau)$ as a signature.

Finally, the vehicle stores (U, κ) on the OBU and sends the query $Q = (N, t_1, U, T, AC', \mathcal{SPK}, c_1, c_2, c_3, \widetilde{B}_1, \widetilde{B}_2, c, \tau)$ to the nearby RSU, if it is in the coverage area of an RSU. Otherwise, the vehicle can send Q to the nearby vehicles, and they deliver the query Q to RSUs via delay-tolerant V2V communications. When the vehicle enters the coverage area of an RSU, it sends Q to the RSU again.

When an RSU with RID receives a query Q from a vehicle, it verifies the validity of the signature $(\widetilde{B}_1, \widetilde{B}_2, c, \tau)$ by computing $B = \hat{e}(\widetilde{B}_1, \widehat{X}^c)\hat{e}(\widetilde{B}_2, \hat{g}^{-c})\ \hat{e}(\widetilde{B}_1, \widehat{X}_1^{\tau})$ and checking whether $c \overset{?}{=} \mathcal{H}(\widetilde{B}_1, \widetilde{B}_2, B, N, t_1, U, T, AC'$, $\mathcal{SPK}, c_1, c_2, c_3)$ holds. If it is invalid, the RSU broadcasts failure and requests the vehicle to re-transmit the query. Otherwise, the RSU checks whether the new query Q has the same tag T with a received query. If yes, it ignores Q. Otherwise, the RSU generates a signature on Q by selecting a random $r_2 \in_R \mathbb{Z}_p$ and computing

$$B_r = g^{r_2}, \quad c_r = \mathcal{H}(RID, Q, B_r), \quad \tau_r = r_2 + zc_r.$$

Finally, the RSU sends (RID, Q, B_r, τ_r) to the cloud server.

When the cloud server receives the query (RID, Q, B_r, τ_r), it verifies the validity of the signature of the RSU by computing $c'_r = \mathcal{H}(RID, Q, B_r)$ and checking the equation $B_r Z^{c'_r} \overset{?}{=} g^{\tau_r}$. If it does not hold, the server returns failure and requests the RSU to re-transmit the query. Otherwise, the server checks whether the tag T in Q is equal to the one in a received query. If yes, the server ignores this query. Otherwise, the server checks the validity of the signature $(\widetilde{B}_1, \widetilde{B}_2, c, \tau)$ and $\mathcal{SPK}$. If either is invalid, the server sends the query Q to the TA and aborts. Otherwise, the server decrypts (c_1, c_2, c_3) to obtain $DEST||CL||AP||t_2||t_3$ as $r_1 = c_2/c_1^{y_1}$, $DEST||CL||AP||t_2||t_3 = AES_{DEC}(r_1, c_3)$. If the query is expired, the server aborts; otherwise, it searches an accessible parking lot for the vehicle according to the destination $DEST$, the current location CL, acceptable price range AP, the expected arrival time t_2 and the real-time data of parking lots.

Response Retrieving. The cloud server firstly generates a navigation response RES, including the location of accessible parking lot, the number of vacant parking spots, the parking price. To prevent the response from being obtained by unregistered vehicles, the server picks two random values $k_1 \in_R \mathbb{Z}_p$, $k_2 \in_R \mathbb{G}_1$ and computes $s_1 = g^{k_1}$, $s_2 = k_2 U^{k_1}$, $s_3 = AES_{ENC}(k_2, RES)$ and $K = U^{y_1}$. Then, to prevent attackers from corrupting the response, the server generates a signature by selecting a random value $k_3 \in_R \mathbb{Z}_p$ to compute $\sigma_1 = g^{k_3}$, $\sigma_2 = \mathcal{H}(t_3, K, s_1, s_2, s_3, \sigma_1)$ and $\sigma_3 = k_3 + y_1\sigma_2$. After that, the cloud server predicts the current location of the vehicle according to the destination and the previous location, and determines $\mathcal{R}$, the set of RSUs that the vehicle would drive through. Finally, the cloud server sends the navigation message $R = (t_3, K, s_1, s_2, s_3, \sigma_1, \sigma_3)$ to the RSUs in $\mathcal{R}$. If the parking information of the recommended parking lot changes, the server generates a new navigation message R^* and sends it to the RSUs in $\mathcal{R}$ in the same way described above.

Upon receiving the message R from the cloud server, each RSU in $\mathcal{R}$ computes $\sigma'_2 = \mathcal{H}(t_3, K, s_1, s_2, s_3, \sigma_1)$ and verifies the signature (σ_1, σ_3) as $\sigma_1 Y_1^{\sigma'_2} \overset{?}{=} g^{\sigma_3}$. If it does not hold, the RSU returns failure and requests the server to re-transmit the navigation message. Otherwise, as shown in Fig. 2, the RSU performs the following steps to store the navigation message:

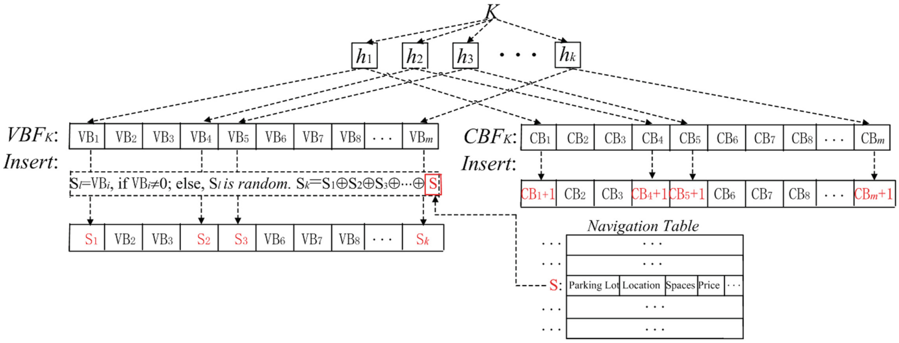


Fig. 2. Insert operation for the RSU.

- Insert K into the counting Bloom filter CBF_K. Specifically, for each $1 \leq l \leq k$, the counter $CB_{h_l(K)}$ increases by one and the rest counters keep the same.
- Store R on the navigation table and obtain the storage address S.
- Insert S into the Bloom filter VBF_K. Firstly, the RSU splits S into k shares of γ-bit, $S_1, S_2, \cdots, S_k$, using the XOR-based secret sharing scheme. If the location on the index $h_l(K)$ of VBF_K has been occupied, the RSU reuses the string $VB_{h_l(K)}$, that is, S_l is fixed to be $VB_{h_l(K)}$, where $l \in \{1, \cdots, k-1\}$; otherwise, S_l is a random γ-bit string. The last string S_k is set to be $S_k = S \oplus S_1 \oplus S_2 \oplus \cdots \oplus S_{k-1}$. Note that the probability that all the locations in VBF_K have been occupied when an address S inserts, is equal to the false positive probability of a Bloom filter. Then, the RSU sets $VB_{h_l(K)}$ to be S_l, for each $1 \leq l \leq k$.

When the vehicle enters the coverage area of an RSU* with RID^* and (z^*, Z^*), it queries whether the parking navigation message R exists on the RSU*. Firstly, the vehicle reads (U, κ) from the memory of the OBU device and computes $K^* = \widehat{Y}_1^{\kappa}$. Then, the vehicle chooses $(u_1, u_2) \in_R \mathbb{Z}_p^2$ to calculate

$$\begin{aligned}(C_1, C_2, C_3) \leftarrow (B_1^{u_1}, B_2^{u_1}, B_3^{u_1 u_2}),\\ \beta_1 = \mathcal{H}(C_1, C_2, C_3, K^*, \tilde{t}),\\ \tau_1 = u_2 + \beta_1 v,\end{aligned}$$

where $\tilde{t}$ is the current time used to resist the replay attack. Finally, the vehicle sends the retrieving query $(K^*, C_1, C_2, \beta_1, \tau_1, \tilde{t})$ to the RSU* to retrieve the response of the navigation query.

Upon receiving $(K^*, C_1, C_2, \beta_1, \tau_1, \tilde{t})$, the RSU* verifies the signature $(C_1, C_2, \beta_1, \tau_1)$ by computing $C_3' = \hat{e}(C_1, \widehat{X}^{\beta_1})\hat{e}(C_2, \hat{g}^{-\beta_1})\hat{e}(C_1, \widehat{X}_1^{\tau_1})$ and checking whether $\beta_1 = \mathcal{H}(C_1, C_2, C_3', K^*, \tilde{t})$ holds. If not, the RSU* returns failure and requests the vehicle to re-send the message. Otherwise, the RSU* checks whether all counters in CBF_K on the locations $h_1(K^*), \cdots h_k(K^*)$ are nonzero. If one of them is zero, the RSU* returns failure to the vehicle and aborts. Otherwise, it recovers the storage address S by computing $S = VB_{h_1(K^*)} \oplus VB_{h_2(K^*)} \oplus \cdots \oplus VB_{h_k(K^*)}$ and finds the navigation message R directly according the storage address S. Then, the RSU* picks a random $r_3 \in_R \mathbb{Z}_p$ to compute $\sigma_1^* = g^{r_3}$, $\sigma_2^* = \mathcal{H}(RID^*, R, \sigma_1^*)$ and $\sigma_3^* = r_3 + z^*\sigma_2^*$. After that, the RSU* returns $(RID^*, R, \sigma_1^*, \sigma_3^*)$ to the vehicle and broadcasts a Bloom filter BF_{K^*} to other RSUs, in which $BF_{K^*}[h_l(K^*)] = 1$ for $1 \leq l \leq k$, and the other bits in the array are zero. Finally, the RSU* performs the deletion operation to remove K^* from CBF_K and delete S in VBF_K. Specifically, the counters in CBF_K on the indices $h_l(K^*)$ for $1 \leq l \leq k$ decrease by one, and the shares of S in VBF_K are removed if the corresponding counters in CBF_K are set to be zero. In addition, if the stored response is expired or an RSU receives a broadcasted BF_{K^*}, the RSU performs deletion operation by deleting the expired or retrieved navigation message and updating the Bloom filters, CBF_K and VBF_K.

If the vehicle receives failure from the RSU*, it can send the retrieving query to other RSUs. Otherwise, the vehicle obtains $(RID^*, R, \sigma_1^*, \sigma_3^*)$. The vehicle checks the validity of the signature (σ_1^*, σ_3^*) by computing $\sigma_4^* = \mathcal{H}(RID^*, R, \sigma_1^*)$ and verifying whether $\sigma_1^*(Z^*)^{\sigma_4^*} = g^{\sigma_3^*}$ holds. If not, the vehicle returns failure and requests the RSU to re-transmit the message. Otherwise, the vehicle calculates $\sigma_4 = \mathcal{H}(t_3, K, s_1, s_2, s_3, \sigma_1)$ and verifies whether $\sigma_1 Y_1^{\sigma_4} = g^{\sigma_3}$ holds. If not, the vehicle sends the message R to the TA for complaint. Otherwise, the vehicle computes $k_2 = s_2/s_1^{\kappa}$ and recovers the navigation response $RES = AES_{DEC}(k_2, s_3)$. Finally, the vehicle can find a vacant parking space according to the parking navigation response. When the vehicle is driving through other RSUs, it would still send the retrieving query to the nearby RSU to check whether the navigation message is updated and retrieve the latest response.

Vehicle Tracing. The vehicle tracing consists of two phases: the cloud server tracing and the TA tracing. In the cloud server tracing phase, the cloud server can recover the identity of a vehicle who submits two different navigation queries in the same time period, which is detected as the denial-of-service attacks. Having two queries Q_1 and Q_2, the cloud server obtains $(DEST, CL, AP, t_1, t_2, t_3, N, T)$ from Q_1 and $(\overline{DEST}, \overline{CL}, \overline{AP}, t_1, \bar{t}_2, \bar{t}_3, \overline{N}, \overline{T})$ from Q_2, respectively. To trace the identity of the vehicle, the server computes $L = \mathcal{H}(DEST, CL, AP, t_1, t_2, t_3, N)$, $\overline{L} = \mathcal{H}(\overline{DEST}, \overline{CL}, \overline{AP}, t_1, \bar{t}_2, \bar{t}_3, \overline{N})$, and $\hat{g}^v = (\frac{T^{\overline{L}}}{\overline{T}^L})^{\frac{1}{t_1(\overline{L}-L)}}$. Then, the cloud server tests $\hat{e}(g, \hat{g}^v) = \hat{e}(V, \hat{g})$ to find the misbehaving vehicle.

In the TA tracing phase, the TA uses the vehicle's signature $(\widetilde{B}_1, \widetilde{B}_2, c, \tau)$ to trace the identity of the vehicle. The TA checks whether $\hat{e}(\widetilde{B}_2, \hat{g}) = \hat{e}(\widetilde{B}_1, \widehat{X})$ $\hat{e}(\widetilde{B}_1, \widehat{V}_1)$ holds or not, until it gets a match.

4 Security Discussion

In this section, we demonstrate that our CPARN meets all security and privacy goals described in Sect. 2.3.

Service Authentication: Each vehicle is delegated with an anonymous credential AC by the cloud server in vehicle registration phase, which is used to access the parking navigation service. To query an available parking spot, the vehicle firstly proves the possession of AC and then sends the navigation query to the cloud server. Therefore, only the vehicles having the anonymous credentials can enjoy this service if the credentials cannot be forged. To generate the credentials for vehicles, the cloud server uses its secret key to sign the commitments of the vehicles to generate a blind signature. Now we show that the unforgeability of the blind signature (A_1, A_3) can be reduced to the modified LRSW Assumption 1 [14]. The credential AC satisfies $A_1 = g^u$, $A_3 = (Yg^tY_1^{VID}Y_2^sY_3^v)^u/g^{ut} = (YY_1^{VID}Y_2^sY_3^v)^u$, which is a valid PS signature on message (VID, s, v). However, the blind signature has the public parameters (Y_1, Y_2, Y_3) compared with the PS signature. Thus, the security of the blind signature can be reduced to the modified LRSW Assumption 1, while the unforgeability of PS signature depends on the modified LRSW Assumption 2 [14]. Therefore, if the modified LRSW Assumption 1 holds, it is impossible for the attackers to forge the anonymous credentials.

Messages Authentication and Integrity: We utilize signature schemes to ensure that all messages sent by authenticated entities cannot be polluted or forged by attackers. The interactions between the cloud server and the RSUs are authenticated using the Schnorr signature scheme, as well as the messages sent by the RSUs to vehicles. Since the Schnorr signature is proved secure under the discrete logarithm assumption, the authentication and integrity of the messages are satisfied. The queries are signed by the vehicles using a randomized secret key, which is a valid PS signature [14] distributed by the TA. Since the PS signature is unforgeable if the modified LRSW Assumption 2 holds, no attacker can forge the secret key (B_1, B_2) and further generate the signatures on vehicles' queries. Therefore, all the exchanged messages between the cloud server and vehicles are authenticated and intact.

Identity Privacy Preservation: We discuss the identity privacy preservation from two aspects. Firstly, in the navigation querying phase, the identities of vehicles cannot be disclosed to the attackers and the curious entities, including the cloud server, RSUs and other vehicles. To prove the possession of the credential AC, the vehicle utilizes the zero-knowledge proof $\mathcal{SPK}$ to show its qualification to enjoy the service, without exposing the identity VID or (V, B_3). The signature on the query $(\widetilde{B}_1, \widetilde{B}_2, c, \tau)$ also does not reveal any information about vehicle's identity, since $\widetilde{B}_1, \widetilde{B}_2$ are randomized and only the TA's public key is required to verify the signature. In addition, although the tag T includes vehicle's secret key v, an attacker cannot identify the vehicle's identity or link two tags to the same vehicle, unless the DDH assumption in $\mathbb{G}_2$ does not hold. Specifically, if there exists an adversary $\mathcal{A}$ that can identify an honest vehicle out of two challenging vehicles, we show how to construct a simulator $\mathcal{S}$ to solve an instance of the Decisional Diffie-Hellman (DDH) problem in $\mathbb{G}_2$. That is, given $G, G_1, G_2, G_3 \in \mathbb{G}_2$, $\mathcal{S}$ can tell whether there exists (ω_1, ω_2), such that $G_1 = G^{\omega_1}, G_2 = G^{\omega_2}, G_3 = G^{\omega_1\omega_2}$.

We use the security model due to Au et al. [15] to formalize the adversary's capacity and the anonymity goal.

$\mathcal{S}$ generates the system parameters and sets $\hat{g} = G$, $\hat{g}_0 = G_1$. $\mathcal{S}$ chooses two vehicles (VID_0, g^{v_0}) and (VID_1, g^{v_1}), where $v_0, v_1 \in_R \mathbb{Z}_p$ and sends them to $\mathcal{A}$. $\mathcal{S}$ simulates the registration phase acting as the authority and the cloud server. $\mathcal{S}$ interacts with $\mathcal{A}$ on behalf of the vehicles VID_0 and VID_1 in the following interactions.

$\mathcal{S}$ honestly acts as VID_0 to answer the parking navigation query. For VID_1, $\mathcal{S}$ randomly picks $\kappa, v, s, t_1, L \in_R \mathbb{Z}_p$ to compute $U = G^{\kappa}$, $T = G^{vt_1}G_1^{Ls}$, generates $(c_1, c_2, c_3, AC', \widetilde{B}_1, \widetilde{B}_2, c, \tau)$, and simulates the zero-knowledge proof $\mathcal{SPK}$ to interact with $\mathcal{A}$.

$\mathcal{S}$ chooses a random $\beta \in \{0, 1\}$. If $\beta = 0$, $\mathcal{S}$ honestly generates a navigation query; otherwise, $\mathcal{S}$ chooses $\kappa^*, v^*, t_1^*, L^* \in_R \mathbb{Z}_p$ to compute $U^* = G^{\kappa^*}$, $T^* = G^{v^*t_1^*}G_3^{L^*}$, and generates $(c_1^*, c_2^*, c_3^*, AC^*, \widetilde{B}_1^*, \widetilde{B}_2^*, c^*, \tau^*)$. $\mathcal{S}$ simulates the zero-knowledge proof $\mathcal{SPK}^*$ and sends them to $\mathcal{A}$. It is easy to see that the simulation is perfect if $log_G G_3 = log_G G_1 \cdot log_G G_2$. Otherwise, it contains no information about VID_0 and VID_1.

Finally, $\mathcal{A}$ returns β'. If $\beta' = \beta$, $\mathcal{S}$ confirms that there exists (ω_1, ω_2), such that $G_1 = G^{\omega_1}, G_2 = G^{\omega_2}, G_3 = G^{\omega_1\omega_2}$. Thus, $\mathcal{S}$ resolves the DDH problem [16] in $\mathbb{G}_2$.

Secondly, in the response retrieving phase, the identities of vehicles are protected against other entities. Specifically, the retrieving query $(K^*, C_1, C_2, \beta_1, \tau_1, \tilde{t})$ sent by the vehicle contains no information about the identity. K^* is a result of Diffie-Hellman agreement, which can be viewed as a random value, and $(C_1, C_2, \beta_1, \tau_1)$ is a signature that only the TA' public key is required for verification. Therefore, our CPARN meets the goal of identity privacy preservation.

Confidentiality: For the navigation queries and responses, we adopt the AES encryption scheme to encrypt them and the Elgamal encryption scheme to securely transmit the symmetric keys to receivers. Specifically, $DEST||CL||AP||t_2||\ t_3$ is protected by a random symmetric key r_1, which is encrypted by the public key of the cloud server to generate (c_1, c_2). Thus, the cloud server can decrypt (c_1, c_2) to obtain the random key r_1 and further recover the navigation query. In terms of the navigation response RES, a random symmetric key k_2 is chosen to encrypt RES and k_2 is encrypted by the vehicle's temporary public key U using the Elgamal encryption scheme. Since the AES encryption and Elgamal encryption are deemed to be secure, the navigation queries and responses are well-protected against the curious vehicles, RSUs and eavesdroppers.

Traceability: The cloud server traces a vehicle's identity successfully if it finds a match of the equation $\hat{e}(g, \hat{g}^v) = \hat{e}(V, \hat{g})$, where $\hat{g}^v = (\frac{T^{\overline{L}}}{\overline{T}^L})^{\frac{1}{t_1(\overline{L}-L)}}$. As the information $(DEST, CL, AP, t_2, t_3)$ is required to compute $\overline{L}$, which can be obtained by decrypting (c_1, c_2, c_3) using the secret key of the cloud server, only the cloud server can trace the identity of the vehicle, who sends more than one navigation queries in a time period. The TA can recover the vehicle's identity by checking the equation $\hat{e}(\widetilde{B}_2, \hat{g}) = \hat{e}(\widetilde{B}_1, \widehat{X})\hat{e}(\widetilde{B}_1, \widehat{V}_1)$. Here $\widehat{V}_1$ is only known by the TA, so that only the TA can recover the vehicle's identity from its signatures.

In summary, CPARN achieves service authentication, message authentication and integrity, identity privacy preservation, confidentiality and traceability, simultaneously.

5 Performance Evaluation

In this section, we evaluate the performance of our CPARN in terms of the computational and communication overheads.

5.1 Computational Overhead

We firstly evaluate the computational overhead of vehicles. By counting the number of the scalar multiplication in $\mathbb{G}_1$ or $\mathbb{G}_2$, AES encryption/decryption, exponentiation in $\mathbb{G}_T$ and bilinear pairing required in each phase, we show the efficiency of CPARN. Other operations, e.g., point addition, integer multiplication, are not resource-consuming compared with the scalar multiplication and bilinear pairing operations. We use T_{SM}, T_{AES}, T_{Exp}, T_p to denote the running time of the scalar multiplication in $\mathbb{G}_1$ or $\mathbb{G}_2$, AES encryption or decryption, exponentiation in $\mathbb{G}_T$ and bilinear pairing operations for vehicles, respectively. We compare our CPARN with VSPN [13] and show the comparison results in Table 1. Since the bilinear pairing operation in querying phase can be precomputed with the aid of the cloud server, there is no bilinear pairing operation in querying and retrieving phases in CPARN, which are frequently performed by the vehicles to enjoy the parking navigation service. The retrieving phase in CPARN is much more efficient than that in VSPN, although the querying phase in CPARN costs a little more time than that in VSPN.

Table 1. Computational burden of vehicles

Phases	CPARN	VSPN
System setup	$8T_{SM}$	$3T_{SM} + T_p + T_{AES}$
Vehicle registration	$13T_{SM} + 3T_p$	$6T_{SM} + T_{AES}$
Navigation querying	$14T_{SM} + 4(T_p) + T_{AES} + 4T_{Exp}$	$T_{SM} + T_{AES}$
Response retrieving	$9T_{SM}$	$4\nu T_p$

$^*\nu$ is the number of RSUs that relay the navigation query in VSPN.

We also run these operations on HUAWEI MT2-L01 smartphone with Kirin 910 CPU and 1250M memory. The operation system is Android 4.2.2 and the toolset is Android NDK r8d with MIRACL 5.6.1 library [17]. The parameter p is approximately 160 bits and the elliptic curve is defined as $y = x^3 + 1$ over

Table 2. Computational burden of RSUs

Phases	CPARN	VSPN
Vehicle registration	0	$2T_p + 3T_{SM} + T_{AES}$
Navigation querying	$3T_p + 4T_{SM}$	$2T_p + T_{SM} + T_{AES}$
Response retrieving	$3T_p + 8T_{SM}$	T_{SM}

$\mathbb{F}_q$, where q is 512 bits. The scalar multiplication operation and AES encryption/decryption operation takes 3.609 ms and 0.023 ms, respectively. The executing time of the exponentiation operation in $\mathbb{G}_T$ and bilinear pairing operation is 0.001 ms and 56.201 ms. Thus, the rough running time of vehicles in system setup and registration phases is 28.869 ms and 215.518 ms, respectively. A vehicle should perform approximately 54.197 ms and 32.478 ms to generate a navigation query and obtain the response.

As for computational overhead of RSUs, we show the comparison results of CPARN and VSPN in Table 2. Our CPARN needs more bilinear pairing operations than VSPN in both querying and retrieving phases. However, these pairing operations in CPARN come from the verification of the vehicle's signature, which is used to ensure the integrity of the messages sent by vehicles, while Chim et al. VSPN [13] does not achieve this security requirement.

5.2 Communication Overhead

We show the communication overhead of CPARN among vehicles, RSUs and the cloud server. The parameters are set the same as those in the simulation. To find a vacant parking space, the vehicle sends the parking navigation query Q, which is $5216 + |N| + |DEST| + |CL| + |AP| + |t_1| + |t_2| + |t_3|$ bits, to the nearby RSU, where $|N|, |DEST|, |CL|, |AP|, |t_1|, |t_2|, |t_3|$ denote the binary length of $N, DEST, CL, AP, t_1, t_2, t_3$, respectively. Then, the RSU appends a 672-bit Schnorr signature to Q and forwards them to the cloud server. The cloud server generates the message R with binary length of $1696 + |t_3| + |RES|$ bits, where $|RES|$ denotes the binary length of RES. After that, the vehicle sends $(K^*, C_1, C_2, \beta_1, \tau_1)$ to the RSU*, which is of the length $1856 + |\tilde{t}|$ bits, where $|\tilde{t}|$ denotes the binary length of $\tilde{t}$. If the navigation message R is stored on RSU*, it returns $(RID^*, R, \sigma_1^*, \sigma_3^*)$ to the vehicle, which is $2368 + |RID^*| + |t_3| + |RES|$ bits, where $|RID^*|$ denotes the binary length of RID^*.

To compare the communication overhead of CPARN and VSPN in the response retrieving phase, we assume the length of navigation response RES in CPARN is equal to that in VSPN and $|RID^*| = |t_3| = 160$ bits. The comparison results are shown in Fig. 3. The communication overhead of vehicles is constant in our CPARN, while the overhead increases linearly with respect to the number of RSUs in VSPN.

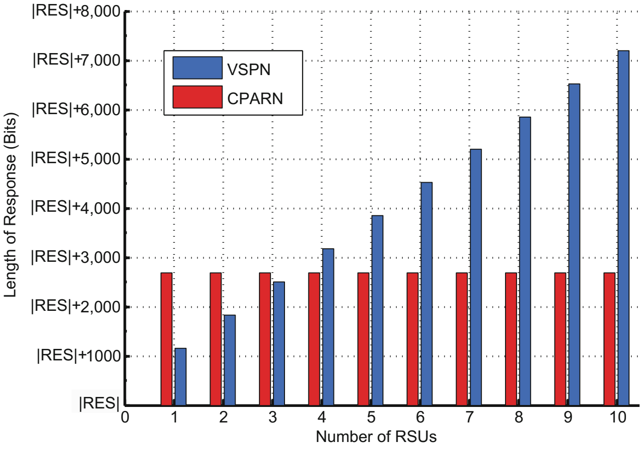


Fig. 3. Communication cost for vehicles.

6 Related Work

Some works [1,13,18,19] have been proposed to achieve privacy-preserving navigation based on VANETs recently. However, the differences between their protocols and ours are significant, as shown in Table 3. Lu et al. [1] presented an intelligent privacy-preserving parking scheme that uses three RSUs to localize the vehicles and assist them to find vacant parking spaces in a large parking lot. While this scheme is of small scale that covers vehicles parking lot. Chim et al. [13] proposed a VANET-based secure and privacy-preserving navigation scheme, in which the online road information collected by RSUs is utilized to guide the drivers to desired destinations in a distributed manner. However, this scheme suffers from inside attack since a system master key is shared among all vehicles. Therefore, Cho et al. [18] developed an improved privacy-preserving navigation protocol to eliminate the system master secret distribution. Consequently, Sur et al. [19] demonstrated that the protocols [13,18] are constructed under the assumption that all RSUs are trusted, and they cannot provide non-transferability of anonymous credentials, i.e., a vehicle can share its credential with others illegitimately. To overcome these weakness, they proposed a secure navigation protocol from one-time credential pseudonymous certificates and proof of knowledge. Different from the existing work, we remove the strong assumption that the querying vehicle can hold the alive connection with the RSU, and allow the vehicle to retrieve the navigation response from the RSUs driving through.

Table 3. Comparison of five navigation protocols

	Lu et al. [1]	Chim et al. [13]	Cho et al. [18]	Sur et al. [19]	CPARN
Privacy preserving	√	√	√	√	√
Cover large scale	X	√	√	√	√
Untrusted RSUs	X	X	X	√	√
Multi-time pseudonym	√	√	√	X	√
No alive connection	X	X	X	X	√

7 Conclusions

In this paper, we have proposed a cloud-based privacy-preserving parking navigation system in VANETs to find accessible parking spots for vehicles. Specifically, a vehicle can query the available parking space to a centralized server and retrieve the result without exposing any sensitive information about the driver. We have presented a novel method to improve the navigation retrieving probability for anonymous vehicular communications under the assumption that the connection between the vehicle and the RSU is difficult to be hold due to the high mobility of the vehicle. Through the security discussion, we have shown that the proposed system meets all the security and privacy goals, and demonstrated its efficiency and practicality for implementation in performance evaluation. For the future work, we will design a privacy-preserving navigation system based on mobile crowdsensing and VANETs to achieve real-time navigation for drivers.

References

1. Lu, R., Lin, X., Zhu, H., Shen, X.: An intelligent secure and privacy-preserving parking scheme through vehicular communications. IEEE Trans. Veh. Technol. **59**(6), 2772–2784 (2010)
2. Vacancy, N.: Park slope's parking problem. http://www.transalt.org/news/releases
3. Thompsona, R.G., Takadab, K., Kobayakawa, S.: Optimisation of parking guidance and information systems display configurations. Transp. Res. Part C Emerg. Tech. **9**(1), 69–85 (2001)
4. Parking guidance and driver information. http://www.ssspl.org/uploads/Products/Pdf/ParkingGuidancesystem.pdf
5. Lin, X., Sun, X., Ho, P.-H., Shen, X.: GSIS: A Secure and Privacy-Preserving Protocol for Vehicular Communications. IEEE Trans. Veh. Technol. **56**(6), 3442–3456 (2007)
6. Lu, R., Lin, X., Luan, T.H., Liang, X., Shen, X.: Pseudonym changing at social spots: an effective strategy for location privacy in VANETs. IEEE Trans. Veh. Technol. **61**(1), 86–96 (2012)
7. Lu, R., Lin, X., Zhu, H., Ho, P.-H., Shen, X.: ECPP: efficient conditional privacy preservation protocol for secure vehicular communications. In: IEEE INFOCOM 2008, Phoenix, AZ, USA, pp. 1903–1911. IEEE Society (2008)
8. Pereira, P.R., Casaca, A., Rodrigues, J.J., Soares, V.N., Triay, J., Cervelló-Pastor, C.: From delay-tolerant networks to vehicular delay-tolerant networks. IEEE Commun. Surv. Tutor. **14**(4), 1166–1182 (2012)

9. Apple, Google collcet user data. http://www.wsj.com/articles
10. How Apple tracks your location without consent, and why it matters. http://arstechnica.com/apple/2011/04
11. Wu, D.J., Zimmerman, J., Planul, J., Mitchell, J.C.: Privacy-preserving shortest path computation. In: NDSS 2016, San Diego, California, USA (2016)
12. Things car thieves know that you do not. http://abcnews.go.com/Business/things-car-thieves/story?id=20938096
13. Chim, T., Yiu, S., Hui, L.C., Li, V.O.: VSPN: VANET-based secure and privacy-preserving navigation. IEEE Trans. Comput. **63**(2), 510–524 (2014)
14. Pointcheval, D., Sanders, O.: Short randomizable signatures. In: Sako, K. (ed.) CT-RSA 2016. LNCS, vol. 9610, pp. 111–126. Springer, Cham (2016). doi:10.1007/978-3-319-29485-8_7
15. Au, M.H., Liu, J.K., Fang, J., Jiang, Z.L., Susilo, W., Zhou, J.: A new payment system for enhancing location privacy of electric vehicles. IEEE Trans. Veh. Technol. **63**(1), 3–17 (2014)
16. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24676-3_4
17. Multiprecision integer and rational arithmetic C/C++ library.http://www.freshports.org/math/miracl/
18. Cho, W., Park, Y., Sur, C., Rhee, K.H.: An improved privacy-preserving navigation protocol in VANETs. J. Wirel. Mob. Netw. Ubiquit. Comput. Dependable Appl. **4**(4), 80–92 (2013)
19. Sur, C., Park, Y., Rhee, K.H.: An efficient and secure navigation protocol based on vehicular cloud. Int. J. Comput. Math **93**(2), 325–344 (2016)