Privacy-preserving Smart Parking Navigation Supporting Efficient Driving Guidance Retrieval

Jianbing Ni, *Student Member, IEEE,* Kuan Zhang, *Member, IEEE,* Yong Yu, *Member, IEEE,*
Xiaodong Lin, *Fellow, IEEE,* Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—It is frustrating and time-consuming for drivers to find an available parking spot in a congested area, such as downtown and shopping malls, especially in peak hours. Thus, it is very helpful for drivers to have real-time parking information to assist them in finding vacant parking spots timely. Unfortunately, to acquire needed parking information, the drivers have to submit personal queries for the availability of parking spaces in their destinations, and this could result in privacy violation if the queries are not protected. To reduce drivers' hassle and preserve drivers' privacy, we propose a privacy-preserving smart parking navigation system (P-SPAN) with efficient navigation result retrieval for drivers using Bloom filters. P-SPAN enables a cloud to guide vehicles to vacant parking spaces in the destinations based on real-time parking information without disclosing any personal information about drivers. Specifically, an efficient data retrieval mechanism is developed based on Bloom filters to support navigation result retrieval for querying vehicles. The drivers can anonymously query accessible parking spots to the cloud, and efficiently retrieve the encrypted navigation results from the passing-by roadside units. Therefore, it is unnecessary for a vehicle to keep connected with the queried roadside unit for acquiring the navigation result. Performance evaluation demonstrates that P-SPAN can provide effective parking navigation with high navigation result retrieving probability and low computational and communication overhead.**

Keywords: Vehicular ad hoc networks (VANETs), smart parking, cloud storage, Bloom filter, security.

I. Introduction

With the increasing number of vehicles in metropolises, finding a vacant parking space in a congested area, such as shopping malls, sport centers and downtown, has become a conflicting and frustrating problem for drivers [2]. It is common for drivers to cruise on road or circle in a parking lot for vacant parking spots. Such vehicles lead to an average 30 percentage of the traffic in crowded areas [3]. This extra traffic gives rise to serious social problems, such as traffic congestion, fuel waste, vehicle accident and air pollution [4]. Although some online navigation systems, e.g., Google Maps and portable navigators, can help drivers to locate parking garages in their desirable destinations, drivers may still common meet that there is no vacant parking space when they arrive, especially in peak hours. Parking guidance information systems [5], [6] have been deployed to broadcast the number of accessible parking spots at some specific spots on roads, but they may increase the traffic pressure around these positions.

Vehicular ad hoc network (VANET) is a particular type of mobile ad hoc network (MANET) where the mobile nodes are vehicles travelling across roads [7]. Each vehicle is equipped with an onboard unit (OBU) to communicate with the nearby vehicles through vehicle-to-vehicle (V2V) communications, and with the roadside units (RSUs) via vehicle-to-roadside (V2R) communications [8], [9]. VANET provides a variety of promising applications to improve road safety and enrich driving experience, in which smart parking navigation offers a real-time parking navigation service to guide on-road vehicles to accessible parking spaces [10]. It is a critical supplement for conventional navigation systems to resist drivers to find vacant parking spots through vehicular communications. In smart parking navigation, the OBU on a vehicle is able to send a parking query to the nearby RUSs for parking space discovery in its destination and reach the accessible parking spot following the up-to-date parking information acquired from the RSUs. It has the advantage that drivers can conveniently use on-board OBUs to access real-time parking navigation services and reach accessible parking spaces within short delay and low fuel cost.

Security and privacy are preliminary concerns for drivers in VANETs [11], since the infrastructure may be confronted with various cyber attacks, including impersonation attacks, forgery attacks and global eavesdropping attacks [12], [13]. To prevent the impersonation attack, it is necessary to authenticate drivers before accessing services, such that a fabricated or unlicensed driver can be detected if it pretends a legal driver to access free services [14]. All messages exchanged between OBUs and RSUs should be signed to prevent the pollution and modification of attackers. The exposure of navigation queries and results is another essential security and privacy problem for smart parking navigation services. A driver is unwilling

Part of this research work was presented in 12th EAI International Conference on Security and Privacy in Communication Networks (SECURECOMM 2016) [1]. This work is supported by the National Key R&D Program of China (2017YFB0802000), NSFC Research Fund for International Young Scientists (61750110528), National Cryptography Development Fund during the 13th Five-year Plan Period (MMJJ20170216), and the Fundamental Research Funds for the Central Universities (GK201702004).

Corresponding Author: Yong Yu

Jianbing Ni and Xuemin (Sherman) Shen are with Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1. Email: {j25ni, sshen}@uwaterloo.ca.

Kuan Zhang is with Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, Omaha, NE 68182 USA, email: kuan.zhang@unl.edu.

Yong Yu is with School of Computer Science, Shaanxi Normal University, Xi'an, 710062, China. Email: yuyong@snnu.edu.cn.

Xiaodong Lin is with Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, Ontario, Canada N2L 3C5, E-mail: xlin@wlu.ca.

to disclose the destination in navigation queries to remain the whereabouts secret, and the protection of navigation results is important to prevent the results sharing with all nearby vehicles if this service is charged; otherwise, the nearby drivers enable to enjoy free parking navigation services, in case they have the same destination with the querying vehicle.

In addition, the leakage of location information is a huge concern for drivers, which triggers numerous controversies on track exposure [15], [16]. Some navigation systems, such as Apple Maps, Google Maps and Baidu Maps, collect drivers' locations and destinations [17], resulting in the leakage of drivers' trajectory and the exposure of their personal habits. In VANET-based parking navigation, OBUs frequently interact with RSUs to deliver personal queries, including current locations and destinations, to acquire real-time parking information. Thus, it is possible for curious entities to learn the driving patterns of vehicles and determine the drivers' locations at a future time, and even identify personal information about drivers, including references, home addresses, workplaces, health conditions, political affiliations and social relationships, according the visiting frequency of specific places. Further, the exposure of vehicles' locations may bring huge convenience to car thieves, as they might trace the vehicles several days before taking action and prefer to steal cars in quiet places [18]. Thereby, location privacy is critical for the wide acceptance of smart parking navigation services to the public. One common method of location privacy preservation is to achieve the anonymity of drivers [19], [20]. Once the drivers are anonymous, no attacker is able to identify the drivers from navigation queries or link several navigation results to reconstruct the trajectory of a specific driver. Unfortunately, after their identities are hidden, how to return navigation results to the target vehicles becomes a new challenge. To resolve this issue without sacrificing drivers' privacy, Chim et al. [21] assume that the vehicle can 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. The handover of V2R connections and signals blocking of buildings increase the disconnection probability of a querying vehicle. As a result, the delivery probability of navigation results is limited. Besides, full anonymity is unrecommended because it is impossible to charge drivers for smart parking navigation services or identify unlicensed or unqualified drivers who get too many demerit points on driving records. Therefore, the drivers' identities should be recovered for service charging and unqualified drivers identification when necessary.

TABLE I
COMPARISON OF FIVE NAVIGATION PROTOCOLS

	[2]	[21]	[22]	[23]	P-SPAN
Privacy Preservation	√	√	√	√	√
Cover Large Scale	X	√	√	√	√
Untrusted RSUs	X	X	X	√	√
Multi-time Pseudonym	√	√	√	X	√
No alive connection	X	X	X	X	√

In this paper, we propose a Privacy-preserving Smart PAking Navigation system (P-SPAN) by integrating vehicular communications and cloud storage to offer secure smart parking navigation services for drivers. We observe that most of drivers use GPS navigation systems, such that the driving route from the source to a destination can be determined. Therefore, the driving-through RSUs for a driver can also be predicted. Thus, drivers can query accessible parking spaces through vehicle communications and acquire the navigation results from the RSUs on the way to the destinations. To be specific, the contributions are three folds:

- P-SPAN enables drivers to query available parking spots by delivering their current locations and destinations to the cloud. The latter searches available parking spots in the destinations based on the real-time parking information. After generating the navigation results, the cloud returns them to the RSUs that the drivers may drive through. Finally, the drivers retrieve the navigation results from the RSUs on the way to their destinations when they enter their coverage areas. With this parking navigation service, the fuels and the time wasted on finding vacant parking spots can be significantly reduced.
- To prevent the privacy leakage of drivers, P-SPAN guarantees conditional privacy preservation for drivers derived from anonymous credentials. In specific, a registered vehicle delivers a personal parking query to the cloud, along with the anonymous credential generated by the cloud, and receives the navigation results without exposing its real identity. At the same time, a trusted authority is able to recover the driver's identity for charging or identifying unqualified drivers.
- We develop an efficient data retrieval mechanism to enhance the retrieving probability of navigation results in anonymous vehicular communications based on Bloom filters. The vehicle can retrieve the navigation result from the RSUs built on the driving routes following GPS navigation information. The probability that vehicles successfully obtain the navigation results can be dramatically improved. This approach is still suitable for the traditional situation where the navigation result is returned rapidly, and the vehicle can receive it from the queried RSU within low latency.

The remainder of this paper is organized as follows. We review the related work in section II and formalize system model, security threats and security goals in section III. In section IV, we propose our P-SPAN system and discuss its security in section V, followed by the discussion on the probability of navigation result retrieval and the performance evaluation in section VI. At last, we draw our conclusion in section VII.

II. Related Work

To support various safety or infotainment applications without exposing drivers' privacy, a variety of privacy-preserving vehicular communication protocols have been proposed in VANETs, including anonymous announcement, secure data forwarding and privacy-preserving traffic monitoring. VANET-based privacy-preserving navigation [2], [21], [22], [23] was introduced to assist vehicles to reach their desired destinations

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2805759, IEEE Transactions on Vehicular Technology

following proper paths with low latency. Lu et al. [2] proposed a privacy-preserving parking scheme for large parking lots, which enables three RSUs to locate vacant parking spaces for vehicles arriving large parking lots. This scheme is of small scale that covers parking lots. Chim et al. [21] presented a VANET-based secure and privacy-preserving navigation scheme, in which the RSUs deployed on roads collect real-time road information and collaboratively guide vehicles to reach the destinations distributively. Unfortunately, this scheme is vulnerable to the internal attack of vehicles as the master key is shared among all vehicles. To resist this attack, Cho et al. [22] proposed an improved privacy-preserving navigation protocol to eliminate the requirement of the master secret key sharing. Consequently, Sur et al. [23] demonstrated that the protocols [21], [22] are designed on strong assumptions that all RSUs are fully trusted and the vehicles would not share their credentials with others illegitimately. To address these weaknesses, Sur et al. proposed a secure navigation protocol from one-time credential and proof of knowledge. However, the schemes [21], [22], [23] depend on the assumption that a moving vehicle is able to finish a query with an RSU, which is quite challenging in reality, particularly, when the vehicle moves at a pretty high speed. Therefore, based on our observation, we remove this assumption and propose a novel smart parking navigation system to achieve parking navigation services for drivers without invading drivers' privacy. Thereby, it is not necessary for the querying vehicle to keep connection alive with the RSU to receive navigation result, instead, the vehicle can retrieve the navigation result from the RSUs on the way to its destination. Such a design is superior to the existing schemes [21], [22], [23], as the retrieving probability of navigation results can be improved. The differences between the schemes in [2], [21], [22], [23] and the P-SPAN are significant, as shown in table I.

III. Problem Statement

We define the problem by formalizing system model and security threats, and identify security goals.

A. System Model

A smart parking navigation system has four entities: a cloud, a large number of vehicles, RSUs and a trusted authority.

- *Cloud*. The cloud, composed of a set of servers and a data center, offers two kinds of services, namely, the parking space management service and smart parking navigation service. In parking space management service, the servers at parking lots and RSUs beside roads collect and manage the status information about parking spots, i.e., possessed, reserved and vacant, charge the parking fee based on the charging policy and outsource the real-time parking data to the data center. The smart parking navigation service for drivers is built on the maintained parking information. For example, the cloud manages the parking lots (red points in Fig.1) around the CN tower and offers smart parking navigation service for the drivers whose desired destinations are CN tower.

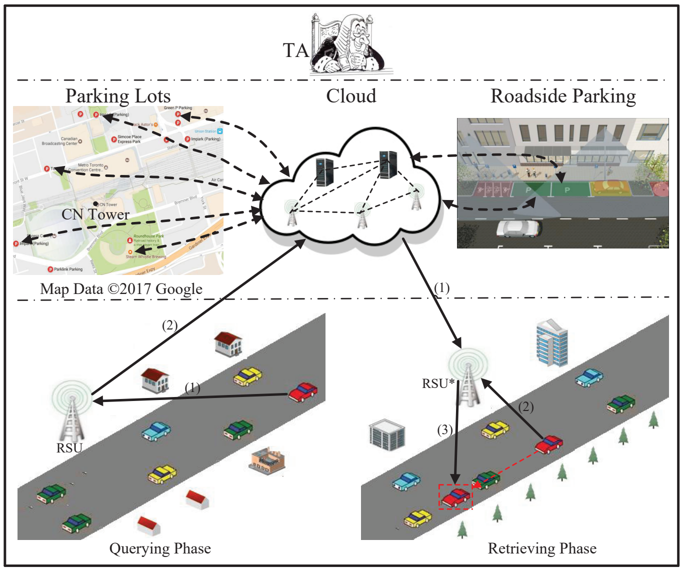


Fig. 1. System Model of P-SPAN

- *Vehicles*. Each vehicle has the capacity to interact with the nearby vehicles and the RSUs using the equipped irreplaceable and temper-proof OBU device. OBUs also have computing capability to execute some simple computations and storage resources to keep data, including a small amount of read-only memory. People with driving licenses can drive vehicles on roads. Demerit points are added to a driver's licence, if the driver is convicted of breaking certain driving laws. If the driver collects enough points, he or she is not qualified to drive in a certain time period.
- *RSUs*. RSUs, deployed on roads, communicate with each other and with driving-through vehicles. They are connected with the Internet to interact with the cloud. Each RSU is resource in rich, indicating that it has sufficient computing capacity to execute cryptographic operations for ensuring the security of information exchange and storage spaces to maintain the navigation results for drivers.
- *Trust Authority (TA)*. TA can be a government agency that administers vehicle registration and driver licensing, such as Department of Motor Vehicles (DMV) in USA and ServiceOntario Centers in Canada. It is a fully trusted authority, whose responsibility is to issue public-key certificates for all the entities in the system, including the cloud, RSUs and vehicles, and recover the drivers' identities in anonymous parking navigation services for service charging or unqualified drivers identification.

Fig. 1 depicts the model of smart parking navigation system. To bootstrap the system, drivers, RSUs and the cloud generate their individual public-secret key pairs to achieve secure communications, TA issues the public-key certificates for all entities, respectively. A driver is required to licence at TA and obtains a digital driving licence after passing driving tests, which is stored on the smart phone or a USB device. The cloud provides parking space management services to parking lots and roadside parking spaces, and maintains real-time parking data outsourced by parking lots through the Internet. To make fully use of the real-time parking data, the cloud offers smart parking navigation services for drivers, which assist drivers to find vacant parking spaces in their destinations. To access this service, a driver is required to register the service at the cloud and acquire an anonymous credential for service access. The smart parking navigation consists of two phases: query and retrieval. In the query phase, (1) a vehicle delivers a parking navigation query generated from its destination and current location, and sends it to the nearby RSU; (2) The RSU receives the parking navigation query and forwards it to the cloud. The cloud searches an available parking space for the querying driver based on the maintained real-time parking information and the driver's desired destination. In the retrieval phase, (1) the cloud predicts the RSUs that the querying vehicle may drive through and returns the navigation result to these RSUs, and the RSUs store the navigation result temporarily; (2) When the querying vehicle enters the coverage area of an RSU, it sends a retrieving query to the RSU; (3) The RSU searches on storage spaces and returns the corresponding navigation result to the vehicle if it maintains that result, otherwise, the vehicle tries to acquire the navigation result from the following RSUs. If the recommended parking space is possessed, the cloud updates the navigation result based on the new location of the vehicle. The vehicle can retrieve the result from the driving-through RSUs and obtain the latest navigation result.

B. Security Threats

Security threats may come from both internal and external attackers. The global eavesdroppers listen on communication channels to capture the transmitting messages exchanged between two entities in the smart parking navigation system, such that it is possible for the eavesdroppers to learn the moving patterns of drivers, guess the locations of drivers at a certain time, and identify personal preferences and habits of drivers from the visiting frequency of points of interest. Internal attackers may be the curious employees in cloud or drivers who are willing to learn more information about other drivers. Although the cloud would follow the regulations and agreements agreed with the drivers, it is also interested in drivers' privacy and eagers to mine private knowledge from the parking navigation queries or learn the driving trajectory of a specific driver. These information, containing numerous privacy about drivers, may be shared with the cooperators for exploiting hidden values. Further, even the cloud behaves honestly on data maintenance, the drivers might still believe that their private information would be revealed to the public, due to the frequently happening accidents of data leakage. Therefore, the cloud is only semi-honest. The vehicles may launch impersonation attacks to pretend legitimate vehicles to enjoy free parking navigation service if this service is charged, or eavesdropping attacks to capture the navigation result, in case they have the same destinations with the querying vehicle. However, they would not share their digital driving licences or anonymous credentials with other vehicles, since they will be punished once discovered by the TA. In addition, the RSUs may be compromised by hackers and they may read the navigation results maintained on storage spaces, or use all sorts of methods to learn sensitive information about drivers by analyzing the forwarding data, e.g., parking queries and navigation results.

C. Security Goals

To achieve privacy-preserving smart parking navigation through vehicular communications under the aforementioned system model and against the security threats, P-SPAN should meet the following security goals:

- Identity Authentication: To ensure that a driver is qualified to drive on roads, indicating that the driver has a digital driving licence and its driving record is good enough for vehicle driving.
- Service Authentication: To guarantee that the vehicle participating in smart parking navigation service is legitimate. It is impossible for an attacker to impersonate a registered vehicle for accessing free navigation service if the service is charged.
- Privacy Preservation: To ensure that the privacy of drivers would not be disclosed in smart parking navigation service. Moreover, given two parking queries, no attacker is able to learn whether both queries are delivered by the same driver, such that the driving pattern of the vehicle is protected.
- Traceability: The TA is able to recover the real identities of the drivers participating in smart parking navigation service for service charging or unqualified drivers identification.

IV. The Proposed P-SPAN

We review the preliminaries and describe P-SPAN in detail.

A. Preliminaries

If S is a non-empty set, $s \in_R S$ denotes s is randomly chosen from S. $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ are cyclic groups with the same prime order p. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is type 3 bilinear pairing [24], 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. Type 1 pairing is that $\mathbb{G}_1 = \mathbb{G}_2$ and type 2 pairing is that $\mathbb{G}_1 \neq \mathbb{G}_2$ and there exists an efficiently computable homomorphism $\pi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$, but there is no efficient homomorphism in the other direction.

Mathematical Assumptions. The secuirty of P-SPAN relies on three mathematical assumptions as follows.

Decisional Diffie-Hellman (DDH) assumption in $\mathbb{G}_2$ [25]. If there is no algorithm can solve the DDH problem, that is, given $(\hat{g}, \hat{g}^a, \hat{g}^b, \hat{g}^c) \in \mathbb{G}_2^4$, to determine $c = ab$ or not, in probabilistic polynomial time with non-negligible probability, then we say that the DDH assumption in $\mathbb{G}_2$ holds.

Modified LRSW assumption 1 [24]. If there is no algorithm can solve the modified LRSW problem 1, that is, given g^b, $\hat{g}^a$, $\hat{g}^b$, where g is a generator of $\mathbb{G}_1$, $\hat{g}$ is a generator of $\mathbb{G}_2$ and $a, b \in_R \mathbb{Z}_p$, and an oracle $\mathcal{O}$, which on input $m \in_R \mathbb{Z}_p$ that chooses a random $h \in_R \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and answers the pair

$P = (h, h^{a+bm})$, to compute a new pair $P' = (h', h'^{a+bm'})$ for $h' \in_R \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and a new m' that is not one of the ms queried in $\mathcal{O}$, in probabilistic polynomial time with non-negligible probability, then we say that the modified LRSW assumption 1 holds.

Modified LRSW assumption 2 [24]. If there is no algorithm can solve the modified LRSW problem 2, that is, given $\hat{g}^a$, $\hat{g}^b$, where $\hat{g}$ is a generator of $\mathbb{G}_2$ and $a, b \in_R \mathbb{Z}_p$, and an oracle $\mathcal{O}$, which on input $m \in_R \mathbb{Z}_p$ that chooses a random $h \in_R \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and answers the pair $P = (h, h^{a+bm})$, to compute a new pair $P' = (h', h'^{a+bm'})$ for $h' \in_R \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and a new m' that is not one of the ms queried in $\mathcal{O}$, in probabilistic polynomial time with non-negligible probability, then we say that the modified LRSW assumption 2 holds.

The modified LRSW assumption 1 and the modified LRSW assumption 2 can be proved to hold in the generic group model.

PS Signature. The PS signature is a public-key signature scheme proposed by Pointcheval and Sanders [24] and its existential unforgeability is proven against chosen message attacks without random oracles under the modified LRSW assumption 2 [24].

Let $\hat{g}$ be a generator of $\mathbb{G}_2$. $(y, x_1, \cdots, x_r) \in_R \mathbb{Z}_p^{r+1}$ is the secret key of the signer and $(\widehat{Y}, \widehat{X}_1, \cdots, \widehat{X}_r) \leftarrow (\hat{g}^y, \hat{g}^{x_1}, \cdots, \hat{g}^{x_r})$ is the public key. A digital signature on multi-block messages $(m_1, \cdots, m_r) \in \mathbb{Z}_p^r$ is $\phi = (\phi_1, \phi_2) = (h, h^{y+\sum_{j=1}^r x_j m_j})$, where h is a random value chosen from $\mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$. The signature ϕ can be publicly verified as $\phi_1 \neq \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and $\hat{e}(\phi_1, \widehat{Y} \prod_{j=1}^r \widehat{X}_j^{m_j}) = \hat{e}(\phi_2, \hat{g})$.

Proof of Knowledge. In a proof-of-knowledge protocol [26], a prover convinces a verifier that she/he possesses a witness w satisfying some type of relation R with respect to a known string x. If the prover is able to convince the verifier in a way that the verifier can learn nothing except the validity of the relation, this protocol is called a zero-knowledge proof-of-knowledge (ZKPoK) protocol [27]. Currently, Σ-protocols, which are a special type of three-move ZKPOK protocols, have been proposed under the honest verifier model. For example, a Σ-protocol that proves the knowledge of discrete logarithm is denoted as $PK\{(x) : y = g^x\}$, indicating that a prover convinces a verifier that she/he possesses $x \in \mathbb{Z}_p$ satisfying $y = g^x$ with respect to some $y \in \mathbb{G}$ without exposing x. Σ-protocols can be transformed into non-interactive Signature Proof-of-Knowledge (SPK) protocols or signature schemes, which can be proven secure under random oracle model. The signature of knowledge for message $m \in \{0,1\}^*$ that is transformed from the above Σ-protocol is denoted as $SPK\{(x) : y = g^x\}(m)$, which is secure under the random oracle model due to Fiat-Shamir heuristic [28].

Bloom Filter [29]. A Bloom filter (BF) is a probabilistic data structure that is used to test whether an element is a member of a set. It uses an array of m bits to represent a set of S with at most n elements and a set of k independent hash functions $H = \{h_1, \cdots, h_k\}$ to uniformly map every element to index numbers over $[0, m-1]$. We use (m, n, k, H)-Bloom filter to denote the Bloom filter with parameters (m, n, k, H), BF_S to denote the Bloom filter that encodes the set S and $BF_S[i]$ to denote the bit on index i in BF_S. Initially, all bits in the array are set to be 0. To insert an element $x \in S$ to the Bloom filter, the hash functions in H are used to map the element x to k index numbers, and the bits at all these k indices in the array are set to be 1, that is, for each $1 \leq l \leq k$, $BF_S[h_l(x)] = 1$. To query an element x' in S, x' is hashed by the hash functions in H to get k index numbers, and then, all locations on k indices in the array should be checked. If one of the bits at these locations is 0, x' does not belong to the set S; otherwise, x' is probably in S.

A counting Bloom filter (CBF) is an extension of the Bloom filter, in which a λ-bit counter replaces the single bit on each index to indicate the number of collisions happened on this location. The CBF offers a method to implement the delete operation on a Bloom filter without recreating the filter. The CBF is denoted as (m, n, k, H, λ)-counting Bloom filter and CB_i denotes the counter on the index i. In the insert operation, the counter CB_i increases if an element is hashed to the index of the counter. In the lookup operation, x is an element of the set S if all counters on the required indices are non-zero. The counters decrease if the element x is deleted from the set S.

B. Overview of P-SPAN

Our P-SPAN consists of five phases: system setup, service registration, parking query, result retrieval and driver tracing. We first provide a high-level description of the P-SPAN, which is designed from the underlying PS signature [24] and Bloom filters [29].

- **System Setup.** The TA setups the whole system by generating system parameters. The TA, the cloud and each RSU generate their secret-public key pairs, respectively. In addition, to acquire a valid digital driving licence, a driver interacts with the TA by executing the ZKPOK protocol derived from the PS signature. The driver commits two values (w, w') and obtains the TA's signature (A_1, A_2, A_3) on (w, w'), while the TA cannot learn anything about (w, w'). The driver obtains its digital driving licence, which consists of the public part (ID, W, A_3) and the secret part $(w, \widehat{W_0}, A_1, A_2)$.
- **Service Registration.** A driver registers the smart parking navigation service on the cloud by executing the ZKPOK protocol derived from the PS signature. The driver makes a commitment and authenticates the identifier W. Upon successful execution of this protocol, the driver obtains the anonymous credential (B_1, B_2) from the cloud.
- **Parking Query.** To find a vacant parking space, the driver encrypts the basic query information and proves its identity by executing SPK protocol without disclosing any information. The driver also randomises (A_1, A_2, A_3) to generate a group signature $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$ on the parking navigation query. The query Q would be sent to the cloud through the relay of the nearby RSU. Finally, the cloud obtains the destination of the driver and finds an available parking lot for the vehicle.
- **Result Retrieval.** The cloud encrypts a navigation result and returns it to the set of RSUs $\mathcal{R}$ predicted to pass by

for the querying driver. Each RSU stores the navigation result on VBF_K to wait the driver to retrieve. If a driver enters a coverage area of an RSU, it generates a retrieving query, which includes a search index K^* and a group signature on K^* to preserve the driver's identity. The RSU searches its VBF_K to retrieve the matched navigation result R if exists; and the driver decrypts to obtain the navigation result.

- Driver Tracing. The TA is able to obtain the digital driving licence of the driver by opening the group signature $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$.

C. The Detailed P-SPAN

The details of P-SPAN are shown as follows.

1) System Setup*:* The TA sets the security parameter ϱ, which denotes the security level of the system. $\varrho = 160$ or 256 in general for Elliptic Curve cryptography. Let p be a large prime with ϱ bits, and ($\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$) be a set of cyclic groups with the same order p. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is the type 3 bilinear pairing. g is a generator of $\mathbb{G}_1$ with $g \neq 1_{\mathbb{G}_1}$, and $\hat{g}, \hat{g}_0$ are two generators of $\mathbb{G}_2$ with $\hat{g} \neq \hat{g}_0 \neq 1_{\mathbb{G}_2}$. $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$ is a collision-resistant secure hash function, $\mathcal{C} = ENC_{AES}(\mathcal{K}, \mathcal{M})$ and $\mathcal{M} = DEC_{AES}(\mathcal{K}, \mathcal{C})$ are the encryption and decryption algorithms of advanced encryption standard (AES), respectively. The TA randomly selects $(y, y_1) \in_R \mathbb{Z}_p^2$ and computes $\widehat{Y} = \hat{g}^y$, $\widehat{Y}_1 = \hat{g}^{y_1}$. (y, y_1) is the secret key of the TA and $(g, \hat{g}, \widehat{Y}, \widehat{Y}_1)$ is the corresponding public key.

The cloud initializes the parking space management service and smart parking navigation service. It manages the status information about parking spaces and uses the real-time parking information to provide smart parking navigation. To generate the secret-public key pair, the cloud randomly chooses $(x, x_1, x_2, x_3) \in_R \mathbb{Z}_p^4$ to calculate

$$(X, X_1, X_2, X_3) \leftarrow (g^x, g^{x_1}, g^{x_2}, g^{x_3}).$$
$$(\widehat{X}, \widehat{X}_1, \widehat{X}_2, \widehat{X}_3) \leftarrow (\hat{g}^x, \hat{g}^{x_1}, \hat{g}^{x_2}, \hat{g}^{x_3}).$$

(x, x_1, x_2, x_3, X) is the secret key of the cloud, and $(X_1, X_2, X_3, \widehat{X}, \widehat{X}_1, \widehat{X}_2, \widehat{X}_3)$ is the corresponding public key.

Each RSU has a unique number RID associated with its location. The RSU randomly chooses $z \in_R \mathbb{Z}_p$ as its secret key and calculates $Z = g^z$ as its public key. The RSU defines two Bloom filters: CBF_K and VBF_K. CBF_K is a (m, n, k, H, λ)-counting Bloom filter and VBF_K is a variant of the traditional Bloom filter. k hash functions $h_l \in H$ in both Bloom filters are defined as $h_l : \mathbb{G}_1 \to \mathbb{Z}_m$, for $1 \leq l \leq k$. VBF_K uses an array of γ-bit strings to indicate the storage addresses of navigation results, rather than an array of bits to represent the set membership in traditional Bloom filter. A 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 counters in CBF_K and the strings in VBF_K are set to be zero.

A driver has a unique identity ID to register at the TA for a digital driving licence after passing driving tests. The driver interacts with the TA in the following steps:

- The driver randomly chooses $(w, w') \in_R \mathbb{Z}_p^2$ to compute $(W, \widehat{W}, \widehat{W'}, \widehat{W_0}) \leftarrow (g^w, \widehat{Y}_1^w \widehat{Y}^{w'}, \hat{g}^{w'}, \widehat{Y}_1^w)$, and sends $(ID, W, \widehat{W}, \widehat{W'})$ to the TA, along with the zero-knowledge proof:
$$\mathcal{PK}_1 = \{(w, w') : W = g^w \wedge \widehat{W} = \widehat{Y}_1^w \widehat{Y}^{w'} \wedge \widehat{W}' = \hat{g}^{w'}\}. \qquad (1)$$
- The TA computes $\widehat{W_1} = \widehat{W} / \widehat{W'}^y$, verifies the proof $\mathcal{PK}_1$ and checks whether the equation $\hat{e}(W, \widehat{Y}_1) = \hat{e}(g, \widehat{W_1})$ holds. If either is invalid, the TA returns failure and aborts. Otherwise, the TA randomly picks $v \in_R \mathbb{Z}_p$ to calculate
$$(A_1, A_2, A_3) \leftarrow (g^v, (g^y W^{y_1})^v, \hat{e}(A_1, \widehat{Y}_1)). \qquad (2)$$
Note that (A_1, A_2) is a PS signature on message w and A_3 is calculated to avoid the bilinear pairing computation for the OBU. Finally, the TA sends (ID, A_1, A_2, A_3) to the driver via a secure channel and keeps $(ID, W, \widehat{W_1})$ secret in a database.
- The driver sets his/her digital driving licence, which consists of two parts, the public part (ID, W, A_3) and the secret part $(w, \widehat{W_0}, A_1, A_2)$. The secret part $(w, \widehat{W_0}, A_1, A_2)$ is kept secretly in a USB device and plugged in the vehicle when the driver starts the vehicle.

2) Service Registration*:* To access the smart parking navigation service, firstly, a vehicle has to register on the cloud to acquire an anonymous credential, which is used to access the service in an anonymous manner. To apply to the credential, OBU on the driver's vehicle interacts with the cloud as follows:

- The OBU randomly selects $(t, s) \in_R \mathbb{Z}_p^2$ to compute $C = g^t X_1^{ID} X_2^s X_3^w$ and the zero-knowledge proof:
$$\mathcal{PK}_2 = \{(t, s, w) : C = g^t X_1^{ID} X_2^s X_3^w \wedge W = g^w\}. \qquad (3)$$
The OBU delivers $(ID, C, W, \mathcal{PK}_2)$ to the cloud.
- The cloud checks the validity of $\mathcal{PK}_2$ and returns failure and aborts if $\mathcal{PK}_2$ is invalid; otherwise, it randomly chooses $u \in_R \mathbb{Z}_p$ to calculate $(B_1, B_2) \leftarrow (g^u, (XC)^u)$. The cloud sends (B_1, B_2) to the OBU via a secure channel and stores (ID, C, W, B_1, B_2) in its database.
- The OBU checks $\hat{e}(B_1, \widehat{X}) \hat{e}(B_1, \hat{g}^t \widehat{X}_1^{ID} \widehat{X}_2^s \widehat{X}_3^w) \overset{?}{=} \hat{e}(B_2, \hat{g})$. If yes, the OBU computes $B_3 = B_2 / B_1^t$, and obtains the anonymous credential $AC = (B_1, B_3)$. At last, it keeps (AC, s) in the read-only memory of the OBU.

3) Parking Query*:* When a driver ID needs the smart parking navigation service, the OBU on the vehicle sends a parking navigation query to the cloud to discover a vacant parking space in the destination. With the digital driving licence and the anonymous credential AC, the OBU generates a parking navigation query as follows:

- Generate the basic query information, including current location CL, the destination DS, current time t_1, expected arrival time t_2, expiration time t_3 and acceptable price range AP.
- Encrypt (DS, CL, AP, t_2, t_3) by randomly choosing $r \in_R \mathbb{Z}_p$, and computing $c_1 = g^r$, $c_2 = \mathcal{H}(c_1, X_1^r)$, and $c_3 = ENC_{AES}(c_2, DS|| \ CL||AP||t_2||t_3)$.

- Randomly pick $\kappa \in_R \mathbb{Z}_p$ to compute a temporary session key $U = \hat{g}^{\kappa}$, $L = \mathcal{H}(ID, DS, AP, t_2, t_3)$ and a tag $T = \hat{g}^{w} \hat{g}_0^{Ls}$.
- Randomly picks $(\alpha, \beta) \in_R \mathbb{Z}_p^2$ to compute $AC' = (B_1', B_3') = (B_1^{\alpha}, (B_3 B_1^{\beta})^{\alpha})$ and generate a zero-knowledge proof as

$$\mathcal{SPK}\left\{\begin{array}{r} (ID, w, s, \kappa, \beta) : \qquad\qquad\qquad\qquad\qquad \\ \hat{e}(B_1', \widehat{X}\hat{g}^{\beta})\hat{e}(B_1', \widehat{X}_1^{ID}\widehat{X}_2^{s}\widehat{X}_3^{w}) = \hat{e}(B_3', \hat{g}) \\ \wedge\, U = \hat{g}^{\kappa} \quad \\ \wedge\, T = \hat{g}^{w}\hat{g}_0^{Ls} \end{array}\right\}(N),$$

  where N is a random number chosen from $\mathbb{Z}_p$ as the identifier of the parking navigation query.
- Randomly choose $(r', r'') \in_R \mathbb{Z}_p^2$ to randomise (A_1, A_2, A_3) by calculating

$$(\widetilde{A}_1, \widetilde{A}_2, \widetilde{A}_3) \leftarrow (A_1^{r'}, A_2^{r'}, A_3^{r'r''}), \tag{4}$$

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

At last, the OBU keeps (U, κ) and sends the parking navigation query $Q = (N, t_1, c_1, c_3, U, T, AC', \mathcal{SPK}, \widetilde{A}_1, \widetilde{A}_2, c, \tau)$ to the nearby RSU, if it is in the coverage area of an RSU. Otherwise, the OBU sends Q the nearby vehicles to reach the nearby RSU via V2V communications. The OBU also temporarily keeps the query Q and delivers Q to an RSU, when the vehicle connects the RSU.

When an RSU with RID receives Q from a vehicle, it first checks whether Q has been received and checks the signature $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$ by calculating $A = \hat{e}(\widetilde{A}_1, \widehat{Y}^{c})\hat{e}(\widetilde{A}_2, \hat{g}^{-c})\ \hat{e}(\widetilde{A}_1, \widehat{Y}_1^{\tau})$ and verifying whether $c \overset{?}{=} \mathcal{H}(\widetilde{A}_1, \widetilde{A}_2, A, N, t_1, U, T, AC', \mathcal{SPK}, c_1, c_3)$ holds. If yes, the RSU verifies whether Q has the same tag T with a received query; otherwise, the RSU returns failure. If the tag T is the same with the tag in a previous query, the RSU ignores Q, otherwise, it randomly selects $r_2 \in_R \mathbb{Z}_p$ to compute a signature on Q by calculating $A_r = g^{r_2}$, $c_r = \mathcal{H}(RID, Q, A_r)$, $\tau_r = r_2 + zc_r$, and forwards (RID, Q, A_r, τ_r) to the cloud.

The cloud verifies the RSU's signature by computing $c_r' = \mathcal{H}(RID, Q, A_r)$ and checking $A_r Z^{c_r'} \overset{?}{=} g^{\tau_r}$ after receiving (RID, Q, A_r, τ_r). If not, the cloud returns failure; otherwise, it verifies whether T in Q is equal to the one in a received query. If yes, the cloud ignores this query; otherwise, it checks the signature $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$ to ensure the validity of ID' driving licence and $\mathcal{SPK}$ to authenticate the validity of the credential AC. If both are valid, the cloud decrypts (c_1, c_3) to obtain $DS||CL||AP||t_2||t_3$ as $c_2' = \mathcal{H}(c_1, c_1^{x_1})$, $DS||CL||AP||t_2||t_3 = DEC_{AES}(c_2', c_3)$. If the query is not expired, the cloud finds an available parking lot for the vehicle based on (DS, CL, AP, t_2) and the real-time parking data of parking lots. In addition, the cloud forwards Q to the TA for charging.

4) Result Retrieval: The cloud first generates a navigation result RS for the parking navigation query, including the geographic location of available parking lot, the quality of accessible parking spots, the parking price and the recommended parking space. The cloud also randomly picks $\psi_1 \in_R \mathbb{Z}_p$ to compute $\phi_1 = g^{\psi_1}$, $\phi_2 = \mathcal{H}(\phi_1, U^{\psi_1})$, $\phi_3 = ENC_{AES}(\phi_2, RS)$ and $K = U^{x_1}$. Then, the cloud generates a signature by randomly choosing $\psi_3 \in_R \mathbb{Z}_p$ to compute $\sigma_1 = g^{\psi_3}$, $\sigma_2 = \mathcal{H}(t_3, K, \phi_1, \phi_3, \sigma_1)$ and $\sigma_3 = \psi_3 + x_1\sigma_2$. After that, the cloud predicts the driving route of the vehicle and determines the set of RSUs $\mathcal{R}$ that the vehicle would pass by. Finally, the cloud forwards the navigation result $R = (t_3, K, \phi_1, \phi_3, \sigma_1, \sigma_3)$ to each RSU in $\mathcal{R}$. If the recommended parking space is possessed by other vehicles, the cloud has to generate a new navigation result R^* for the querying vehicle and forwards it to the RSUs in $\mathcal{R}$.

Upon receiving R, each RSU in $\mathcal{R}$ calculates $\sigma_2' = \mathcal{H}(t_3, K, \phi_1, \phi_3, \sigma_1)$ and checks (σ_1, σ_3) as $\sigma_1 X_1^{\sigma_2'} \overset{?}{=} g^{\sigma_3}$. The RSU returns failure if the equation does not hold; otherwise, the RSU keeps R in the following steps, as shown in Fig. 2:

- Insert K into CBF_K. Specifically, the counter $CB_{h_l(K)}$ increases by one and the rest counters keep the same for each $1 \leq l \leq k$.
- Keep R on the memory and acquire the storage address S.
- Insert S into VBF_K. Specifically, the RSU divides 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 $h_l(K)$ of VBF_K has been occupied, the RSU reuses the string $VB_{h_l(K)}$, i.e., S_l is set to be $VB_{h_l(K)}$, in which $l \in \{1, \cdots, k-1\}$; otherwise, S_l is fixed to be a random γ-bit string. The last string S_k is computed as $S_k = S \oplus S_1 \oplus S_2 \oplus \cdots \oplus S_{k-1}$, if $VB_{h_k(K)} = 0$; otherwise, find an unpossessed location on $h_l(K)$ to set $S_l = S \oplus S_1 \oplus \cdots \oplus S_{l-1} \oplus S_{l+1} \oplus \cdots \oplus S_k$.

When the vehicle enters the coverage area of an RSU* (with an identifier RID^* and a secret-public key pair (z^*, Z^*)), it checks whether the parking navigation result R is maintained on the RSU* and retrieves the result from RSU*. The OBU first obtains (U, κ) from its memory and calculates $K^* = \widehat{X}_1^{\kappa}$. Then, the OBU randomly picks $(u_1, u_2) \in_R \mathbb{Z}_p^2$ to compute a signature as

$$\begin{aligned} (C_1, C_2, C_3) &\leftarrow (A_1^{u_1}, A_2^{u_1}, A_3^{u_1 u_2}), \\ \beta_1 &= \mathcal{H}(C_1, C_2, C_3, K^*, \tilde{t}), \\ \tau_1 &= u_2 + \beta_1 w, \end{aligned}$$

where $\tilde{t}$ is the timestamp. Finally, the OBU forwards the retrieving query $(K^*, C_1, C_2, \beta_1, \tau_1, \tilde{t})$ to the RSU* for navigation result retrieval.

Upon receiving the retrieving query from the OBU, the RSU* performs the following steps to find the corresponding navigation result.

- Verify the signature $(C_1, C_2, \beta_1, \tau_1)$ by calculating $C_3' = \hat{e}(C_1, \widehat{Y}^{\beta_1})\hat{e}(C_2, \hat{g}^{-\beta_1})\hat{e}(C_1, \widehat{Y}_1^{\tau_1})$ and verifying whether $\beta_1 = \mathcal{H}(C_1, C_2, C_3', K^*, \tilde{t})$ or not, and return failure and abort if the equation does not hold.
- Check whether the counters on the locations $(h_1(K^*), \cdots h_k(K^*))$ in CBF_K are nonzero, and return failure and abort if one of the counters is zero.
- Recover the storage address S as $S = VB_{h_1(K^*)} \oplus VB_{h_2(K^*)} \oplus \cdots \oplus VB_{h_k(K^*)}$ and find the navigation result R on the storage address S.
- Randomly choose $r_3 \in_R \mathbb{Z}_p$ to compute $\sigma_1^* = g^{r_3}$, $\sigma_2^* =$

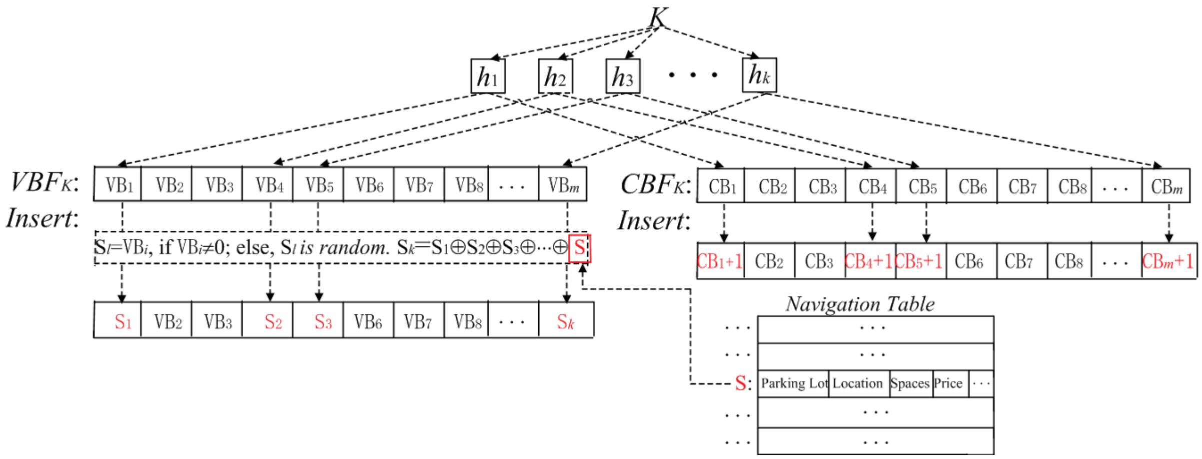

K
h1
h2
h3
• • •
hk
VBFK:
VB1 VB2 VB3 VB4 VB5 VB6 VB7 VB8 · · · VBm
Insert:
Sl=VBi, if VBi≠0; else, Sl is random. Sk=S1⊕S2⊕S3⊕···⊕ S
S1 VB2 VB3 S2 S3 VB6 VB7 VB8 · · · Sk
CBFK:
CB1 CB2 CB3 CB4 CB5 CB6 CB7 CB8 · · · CBm
Insert:
CB1+1 CB2 CB3 CB4+1 CB5+1 CB6 CB7 CB8 · · · CBm+1
Navigation Table
S:
Parking Lot
Location
Spaces
Price

Fig. 2. Insert Operation for the RSU.

$\mathcal{H}(RID^*, R, \sigma_1^*)$ and $\sigma_3^* = r_3 + z^*\sigma_2^*$.

The RSU* sends $(RID^*, R, \sigma_1^*, \sigma_3^*)$ to the OBU and performs the deletion operation to remove K^* from CBF_K and S from VBF_K. In specific, 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 navigation result is expired, the RSU also performs deletion operation to update CBF_K and VBF_K.

After obtaining $(RID^*, R, \sigma_1^*, \sigma_3^*)$, the OBU computes $\sigma_4^* = \mathcal{H}(RID^*, R, \sigma_1^*)$ and checks whether $\sigma_1^*(Z^*)^{\sigma_4^*} = g^{\sigma_3^*}$. If not, the OBU returns failure; otherwise, it computes $\sigma_4 = \mathcal{H}(t_3, K, \phi_1, \phi_3, \sigma_1)$ and verifies whether $\sigma_1 X_1^{\sigma_4} = g^{\sigma_3}$. If not, the OBU forwards R to the TA for complaint; otherwise, the OBU calculates $\phi_2' = \mathcal{H}(\phi_1, \phi_1^{\kappa})$ and recovers the navigation result $RS = DEC_{AES}(\phi_2', \phi_3)$. According to this information, the driver can find a vacant parking space in the destination. When the vehicle drives through other RSUs, it would till send the retrieving query to the nearby RSU to check whether the navigation result is updated and retrieve the latest one.

5) Driver Tracing: The TA is able to know the digital driving licence of the driver from the signature $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$. Specifically, the TA uses each $(ID, W, \widehat{W}_1)$ to test whether $\hat{e}(\widetilde{A}_2, \hat{g}) = \hat{e}(\widetilde{A}_1, \widehat{Y})\ \hat{e}(\widetilde{A}_1, \widehat{W}_1)$ holds or not, until it gets a match.

V. Security Discussion

We discuss the security properties of our proposed P-SPAN, including identity authentication, service authentication, privacy preservation and traceability.

Identity Authentication: The driver with an eligible digital driving licence is qualified to drive on roads. The digital driving licence is generated based on the PS signature interacting with the TA, who is responsible for issuing driving licences to drivers. The TA generates the driving licence $(ID, W, w, \widehat{W}_0, A_1, A_2, A_3)$ and delegates it to the driver in system setup phase, which is randomized in parking query phase to compute the signature on the parking query $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$. Through the signature verification, the verifier can learn whether the driver is eligible for driving or not. Since the signature on parking query is generated from the driving licence $(ID, W, w, \widehat{W}_0, A_1, A_2, A_3)$, only the driver having an eligible driving licence is able to compute a valid signature $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$. Therefore, the identity authentication of drivers depends on the driving licence, which is a PS signature generated by the TA. Since the PS signature is unforgeable based on the modified LRSW assumption 2, no attacker can forge the driving licence $(ID, W, w, \widehat{W}_0, A_1, A_2, A_3)$ and further generate the signatures on navigation queries. Therefore, the drivers are authenticated to ensure that only drivers with eligible driving licences can drive on roads.

Service Authentication: In service registration phase, the OBU on each vehicle interacts with the cloud to generate an anonymous credential AC, which is used to access the smart parking navigation service. To query a vacant parking space, the OBU should prove the possession of AC using zero-knowledge proof to the cloud to show the access capability of parking navigation service. Therefore, only the vehicles with valid anonymous credentials can enjoy this service. To generate an anonymous credential AC for a vehicle, the cloud signs the commitment C using its secret key to compute a signature (B_1, B_2) and the vehicle generates $AC = (B_1, B_3)$ from (B_1, B_2). The unforgeability of anonymous credential (B_1, B_3) can be reduced to the modified LRSW assumption 1 [24]. The anonymous credential AC satisfies $B_1 = g^u$, $B_3 = (Xg^t X_1^{ID} X_2^s X_3^w)^u / g^{ut} = (X X_1^{ID} X_2^s X_3^w)^u$, which is a valid PS signature on (ID, s, w). The public parameters are (X_1, X_2, X_3) compared with the PS signature. Therefore, the security of anonymous credential can be reduced to the modified LRSW assumption 1, while the unforgeability of PS signature depends on the modified LRSW assumption 2 [24]. In short, if the modified LRSW assumption 1 holds, no attacker can forge an anonymous credential to access smart parking navigation service.

Privacy Preservation: To show the privacy preservation of drivers in P-SPAN, we prove that the driver's identity would not be disclosed in parking query and result retrieval phases. Firstly, in parking query phase, the driver delivers a parking navigation query $Q =$

$(N, t_1, c_1, c_3, U, T, AC', \mathcal{SPK}, \widetilde{A}_1, \widetilde{A}_2, c, \tau)$ to the cloud, in which the service authentication message $(AC', \mathcal{SPK})$, the signature $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$ and the tag T are associated with the driver's identity. The service authentication message $(AC', \mathcal{SPK})$ would not disclose the driver's identity since $AC' = (B_1', B_3')$ is randomized from (B_1, B_3) using randomly chosen values (α, β) and the zero-knowledge proof $\mathcal{SPK}$ is sound. Thus, it is impossible to distinguish (B_1', B_3') with random values or link the service authentication messages to a specific driver. The signature $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$ would not expose any personal information about drivers, after (A_1, A_2, A_3) are randomized by random values (r', r'') and only the public key of TA is used to check the validity of the signature $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$. Although the tag T has the driver's secret value w, it is impossible for an attacker to identify the driver's identity or link two tags to the same driver, unless the Decisional Diffie-Hellman problem (DDH) in $\mathbb{G}_2$ [25] is easy to solve. We claim that if an adversary $\mathcal{A}$ is able to identify an honest driver out of two challenging identities, there exists a simulator $\mathcal{S}$ to solve an instance of the DDH problem in $\mathbb{G}_2$, that is, given $(H, H_1, H_2, H_3) \in \mathbb{G}_2^4$, $\mathcal{S}$ can say whether there exists (ω_1, ω_2), such that $H_1 = H^{\omega_1}, H_2 = H^{\omega_2}, H_3 = H^{\omega_1 \omega_2}$. The security model of the identity privacy preservation is defined in [30] for the formalization of the adversary's capacity and the anonymity goal.

$\mathcal{S}$ setups the system parameters and sets $\hat{g} = H$, $\hat{g}_0 = H_1$. $\mathcal{S}$ picks two drivers' identities (ID_0, g^{w_0}) and (ID_1, g^{w_1}), in which $(w_0, w_1) \in_R \mathbb{Z}_p^2$ and forwards them to $\mathcal{A}$. $\mathcal{S}$ simulates the system setup and service registration phases acting as the TA and the cloud. $\mathcal{S}$ also interacts with $\mathcal{A}$ on behalf of the drivers ID_0 and ID_1 in the following interactions.

$\mathcal{S}$ honestly answer parking navigation queries acting as ID_0. For ID_1, $\mathcal{S}$ picks random values $(\kappa, w, s, t_1, L) \in_R \mathbb{Z}_p^5$ to calculate $U = H^{\kappa}$, $T = H^w H_1^{Ls}$, generates $(c_1, c_3, AC', \widetilde{A}_1, \widetilde{A}_2, c, \tau)$, and simulates the zero-knowledge proof $\mathcal{SPK}$ to interact with $\mathcal{A}$.

$\mathcal{S}$ randomly picks $\beta \in \{0, 1\}$. If $\beta = 0$, $\mathcal{S}$ honestly generates a parking navigation query; otherwise, $\mathcal{S}$ randomly picks $(\kappa^*, w^*, t_1^*, L^*) \in_R \mathbb{Z}_p^4$ to calculate $U^* = H^{\kappa^*}$, $T^* = H^{w^*} H_3^{L^*}$, and generates $(c_1^*, c_3^*, AC^*, \widetilde{A}_1^*, \widetilde{A}_2^*, c^*, \tau^*)$. $\mathcal{S}$ simulates the zero-knowledge proof $\mathcal{SPK}^*$ and forwards them to $\mathcal{A}$. It is easy to see that the game is perfectly simulated by $\mathcal{S}$ if $log_H H_3 = log_H H_1 \cdot log_H H_2$. Otherwise, $\mathcal{S}$ cannot contain any information about ID_0 and ID_1.

At last, $\mathcal{A}$ returns β'. If $\beta' = \beta$, $\mathcal{S}$ confirms that there exists (ω_1, ω_2), such that $H_1 = H^{\omega_1}, H_2 = H^{\omega_2}, H_3 = H^{\omega_1 \omega_2}$. Therefore, $\mathcal{S}$ addresses the DDH problem in $\mathbb{G}_2$.

Secondly, in result retrieving phase, the driver's identity is well preserved against the attackers, since the retrieving query $(K^*, C_1, C_2, \beta_1, \tau_1, \ \tilde{t})$ would not expose any information about the driver. 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 randomized from (A_1, A_2, A_3) and only the TA' public key is required for verification. Therefore, the driver's identity is not exposed in result retrieval phase.

Traceability: To recover the driver's identity, the TA uses the maintained $(ID, W, \widehat{W}_1)$ to check the equation $\hat{e}(\widetilde{A}_2, \hat{g}) = \hat{e}(\widetilde{A}_1, \widehat{Y})\hat{e}(\widetilde{A}_1, \widehat{W}_1)$, until finding a matched $(ID, W, \widehat{W}_1)$. Since $\widehat{W}_1$ is kept by the TA, only the TA can recover the vehicle's identity from $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$.

In summary, P-SPAN achieves identity authentication, service authentication, privacy preservation and traceability.

VI. Retrieving Probability and Performance Analysis

We discuss the retrieving probability of navigation results for drivers and evaluate the computational, communication and storage overhead of P-SPAN.

A. Retrieving Probability

To ensure the retrievability of navigation results, we use a counting Bloom filter CBF_K to count the number of collisions occur on each index, and a variant of Bloom filter VBF_K to indicate the storage addresses of navigation results on the RSU. Because of the false positive probability of Bloom filter, drivers probably retrieve false navigation results, which means that K^* does not exist in CBF_K, but all $CB_{h_l(K^*)}$ are set to be non-zero, for $1 \leq l \leq k$. In CBF_K, the probability that a counter is set to be non-zero is $P = 1 - (1 - \frac{1}{m})^{kn}$. Thus, the upper bound of false positive probability is

$$\epsilon = (1 + O(\frac{k}{P}\sqrt{\frac{\mathrm{In}m - k\mathrm{In}P}{m}}))P^k, \qquad (5)$$

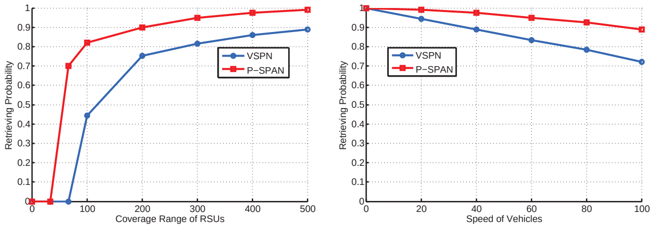
which is negligible in k. In VBF_K, the false positive probability is also ϵ since the shares of storage address S replace the counters in CBF_K from the high level point of view. Therefore, the low bound of retrieving probability is $1 - \epsilon$. In fact, the probability to retrieve a false navigation result is much smaller than ϵ, because the shares on the indices $h_1(K^*), \cdots, h_k(K^*)$ in VBF_K may not consist of a correct storage address, if K^* is not an element in CBF_K. If we require the probability of successful retrieval to be at least θ, the lower bound of m is $m > nlog_2 e \cdot log_2 1/(1 - \theta)$, where e is the base of national logarithms.

The above analysis demonstrates that a driver is able to obtain the correct navigation result from an RSU in result retrieval phase if the required result is stored on the RSU. Now we assume the probability that the j-th driving-through RSU is storing the navigation result, when the driver submits its retrieving query, is ϕ_j. The first RSU is the one that forwards the parking query for the driver, and the driver would drive through ν RSUs before it arrives its desired destination. Thus, the probability that the driver obtains the navigation result from the j-th RSU is

$$\prod_{i=1}^{j-1}(1 - \phi_i)(1 - \theta)^{j-1}\phi_j, \qquad (6)$$

and the probability that the driver receives the navigation result before it arrives the destination is

$$\sum_{j=1}^{\nu}(\prod_{i=1}^{j-1}(1 - \phi_i)(1 - \theta)^{j-1}\phi_j). \qquad (7)$$



(a) The Range of RSUs.

(b) The Speed of Vehicles.

Fig. 3. Comparison on Retrieving Probability.

Compared with the existing protocols [21], [22], [23], which can only receive the result from the first RSU, P-SPAN can significantly improve the navigation retrieving probability.

We simulate the retrieving probability of navigation results returned from the cloud. According to the testing result in [31], a vehicle needs about 1 second to build a connection with an RSU if the packet lost probability is 50% due to the poor channel condition and 15 vehicles wait in the queue to access the RSU. The maximum coverage range of an RSU is 500 meters, and the delay of smart parking navigation query is around 2 seconds considering the queueing delay, transmission delay in two-round interactions and the processing delay for the cloud. The OBU generates parking navigation queries, sends to the nearby vehicles or RSUs at anytime and waits to the navigation results in VSPN. In P-SPAN, the OBU receives the results from the querying RSUs if the connection is kept, otherwise, it retrieves the results from new RSUs when passing by. The simulation result is shown in Fig. 3. With the increasing RSUs' coverage range, the retrieving probability of navigation results increase significantly for both VSPN and P-SPAN. Our P-SPAN can achieve higher retrieving probability than VSPN in Fig. 3(a), since the OBUs can retrieve the navigation results from the passing-by RSUs. If the speed of vehicle increases, the retrieving probability would be decreased, but P-SPAN still has higher retrieving probability compared with VSPN, as shown in Fig. 3(b).

B. Collision Reduction in VBF_K

When the RSU adds a storage address S to the Bloom filter VBF_K, we should ensure that at least one of the indices K hashes to does not been occupied by the previously added addresses. Now we analyze the probability that the RSU can insert n addresses into VBF_K successfully. Assume that the RSU has added $n-1$ addresses successfully, the probability that a particular position is occupied is at most $P' = 1-(1-\frac{1}{m})^{k(n-1)}$. When inserting the n-th address, the probability that all k positions have been occupied is at most

$$\epsilon' = (1+O(\frac{k}{P'}\sqrt{\frac{\text{In}m - k\text{In}P'}{m}}))P'^k, \quad (8)$$

where $P' = 1-(1-1/m)^{k(n-1)}$. Therefore, the probability of inserting n storage addresses into VBF_K is $1-\epsilon'$, which is the false positive probability of a Bloom filter.

If all the locations in VBF_K have been occupied when an address S inserts, one trivial method is to build another Bloom filter VBF'_K to keep S. However, this approach wastes storage spaces for the RSU. To improve the storage efficiency and reduce the collision, we extend the Bloom filter VBF_K to guarantee that a storage address in RSU can be inserted into VBF_K smoothly. Specifically, we employ $k+k'$ hash functions to setup the Bbloom filter VBF_K, namely, $(h_1, \cdots, h_k, h_{k+1}, h_{k+2}, \cdots, h_{k+k'})$ and build an element below each array in VBF_K, as shown in Fig 4. When the RSU inserts the storage address S into VBF_K, it checks whether all the locations on the indices $h_l(K)$ in VBF_K have been occupied for $l \in \{1, \cdots, k\}$. If not, the RSU splits S into $S_1, S_2, \cdots, S_k$ using the XOR-based secret sharing scheme and inserts them into VBF_K following the method described in result retrieval phase of P-SPAN; otherwise, the RSU computes $h_{k+1}(K)$ and checks whether the location on the index $h_{k+1}(K)$ is occupied or not. If not, the RSU fixes S_l to be $VB_{h_l(K)}$, for $l \in \{1, \cdots, k\}$, computes $S_{k+1} = S \oplus S_1 \oplus S_2 \oplus \cdots \oplus S_k$ and sets S_{k+1} below $VB_{h_{k+1}(K)}$; otherwise, the location on $h_{k+1}(K)$ has been occupied and the RSU computes $h_{k+2}(K)$ to find the position below the array $VB_{h_{k+2}(K)}$ to keep the share S_{k+2}, if the location on the index $h_{k+2}(K)$ is not occupied, until the last hash function $h_{k+k'}$ is leveraged. If all extended hash functions $(h_{k+1}, h_{k+2}, \cdots, h_{k+k'})$ are utilized and all the locations on indices $h_1(K), \cdots, h_k(K), h_{k+1}(K), \cdots, h_{k+k'}(K)$ are occupied in VBF_K, the RSU has to build another Bloom filter VBF'_K to maintain the storage addresses. To retrieve the navigation result, the RSU* computes the storage address S as $S = VB_{h_1(K^*)} \oplus VB_{h_2(K^*)} \oplus \cdots \oplus VB_{h_k(K^*)}$, if the element below the location on the index $h_{k+1}(K^*)$ is vacant; otherwise, it should check whether there is any element below the location on the index $h_{k+2}(K^*)$. If not, the RSU^* computes S as $S = VB_{h_1(K^*)} \oplus VB_{h_2(K^*)} \oplus \cdots \oplus VB_{h_k(K^*)} \oplus VB_{h_{k+1}(K^*)}$; otherwise, it further checks the element below the location on the index $h_{k+3}(K^*)$ to recover the storage address S until the location on $h_{k+k'}(K^*)$ is checked.

In the extension, the probability that a particular position is occupied is at most $P^* = 1-(1-\frac{1}{m})^{(k+k')(n-1)}$. When inserting the n-th address, the probability that all $k+k'$ positions have been occupied is at most

$$\epsilon^* = (1+O(\frac{k+k'}{P^*}\sqrt{\frac{\text{In}m - (k+k')\text{In}P^*}{m}}))P^{*k+k'}, \quad (9)$$

where $P^* = 1-(1-1/m)^{(k+k')(n-1)}$. Therefore, the probability of inserting n storage addresses into VBF_K is $1-\epsilon^*$.

C. Computational Overhead

We evaluate the computational overhead of P-SPAN by counting the number of time-consuming cryptographic operations in each phase, including scalar multiplication in $\mathbb{G}_1/\mathbb{G}_2$, AES encryption/decryption, exponentiation in $\mathbb{G}_T$ and bilinear pairing. Other operations, such as point addition, integer multiplication and hash function, are not resource-consuming compared with scalar multiplication and bilinear pairing. We use T_{SM}, T_{AES}, T_{Exp} and T_p to denote the running time of scalar multiplication in $\mathbb{G}_1/\mathbb{G}_2$, AES encryption/decryption, exponentiation in $\mathbb{G}_T$ and bilinear pairing for vehicles, respectively. To demonstrate the high efficiency of our P-SPAN, we

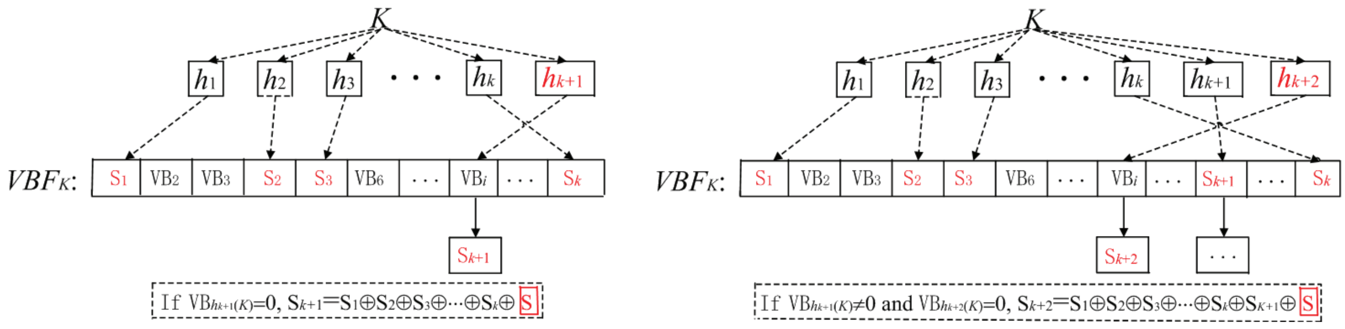


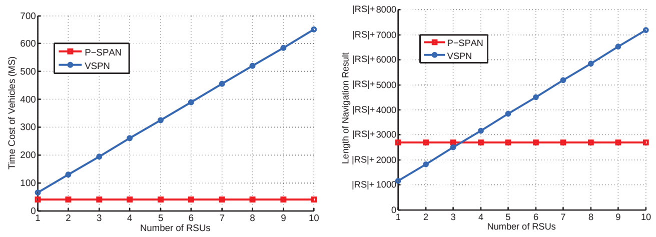
Fig. 4. Extension of VBF_K.

TABLE II
COMPUTATIONAL OVERHEAD OF VEHICLES

Phases	P-SPAN	VSPN
System Setup	$8T_{SM}$	$3T_{SM}$+T_p+T_{AES}
Vehicle Registration	$13T_{SM}$+$3T_p$	$6T_{SM} + T_{AES}$
Parking Query	$14T_{SM}$+4(T_p)+T_{AES} +$4T_{Exp}$	T_{SM}+T_{AES}
Result Retrieval	$9T_{SM}$	$4\nu T_p$

compare P-SPAN with VSPN [21] and show the comparison results in Table II. Although P-SPAN is less efficient than VSPN in service registration phase, this phase is executed only once for each vehicle. In parking query phase, four bilinear pairings $\hat{e}(B_1, \widehat{X})$, $\hat{e}(B_1, \widehat{X}_1)$, $\hat{e}(B_1, \widehat{X}_2)$ and $\hat{e}(B_1, \widehat{X}_3)$ can be pre-computed in service registration phase with the aid of the cloud. Thus, no bilinear pairing is executed in parking query and result retrieval phases of P-SPAN. Furthermore, P-SPAN is more efficient than VSPN in result retrieval phase, since VSPN requires each OBU to perform 4ν bilinear pairings to retrieve the navigation result from RSUs, where ν is the number of RSUs to generate the navigation result for the driver.

P-SPAN is a VANET-based smart parking navigation system implementable on OBUs, RSUs and the cloud, which brings huge convenience to drivers on parking space discovery. To evaluate the practicality of P-SPAN, we execute our P-SPAN on a notebook with Intel Core i5-4200U CPU @2.29GHz and 4.00GB memory. We use MIRACL library 5.6.1 [32] to implement number-theoretic based methods of cryptography. The R-ATE pairing [33] is utilized to realize the bilinear pairing. To ensure the security of P-SPAN, the parameter p is approximately 160 bits. The execution time of OBU in system setup and service registration phases is 30.376 ms and 143.649 ms, respectively. The OBU has to execute approximately 62.535 ms and 38.284 ms to deliver a parking navigation query and retrieve the navigation result. Therefore, P-SPAN is computation-efficient to be implemented on OBUs. Fig. 5(a) shows the comparison result between P-SPAN and VSPN about the time cost of OBU in result retrieval phase. The computational overhead of OBU in result retrieval phase of P-SPAN is constant and pretty low, while the executing time of OBU to read the navigation result in VSPN is linear with the number of RSUs participating in the navigation result generation.



(a) Time Cost.

(b) Communication Cost.

Fig. 5. Performance Comparison for OBUs on Result Retrieval.

D. Communication Overhead

To demonstrate the communication overhead of P-SPAN, we count the length of exchanged messages among vehicles, RSUs and the cloud. The system parameters are set to be the same as those in the simulation, in which $\varrho = 160$. In each parking query phase, the OBU needs to deliver a smart parking navigation query Q to the nearby RSU, which is 5216+$|N| + |DS| + |CL| + |AP| + |t_1| + |t_2| + |t_3|$ bits, where $|N|, |DS|, |CL|, |AP|, |t_1|, |t_2|, |t_3|$ are the binary length of $N, DS, CL, AP, t_1, t_2, t_3$, respectively. The RSU verifies the signature $(\widetilde{A}_1, \widetilde{A}_2, c, \tau)$, appends a 672-bit Schnorr signature (A_r, τ_r) to Q and forwards (A_r, τ_r, Q) to the cloud. Upon receiving the parking query, the cloud generates the navigation result R with 1696+$|t_3| + |RS|$ bits and sends R to the RSUs that the querying vehicle may drive through, where $|RS|$ is the binary length of RS. When the vehicle enters the coverage area of an RSU*, it sends 1856+$|\tilde{t}|$-bit $(K^*, C_1, C_2, \beta_1, \tau_1)$ to the RSU*, where $|\tilde{t}|$ is the binary length of $\tilde{t}$. If R is maintained on RSU*, RSU* returns $(RID^*, R, \sigma_1^*, \sigma_3^*)$ to the querying vehicle, which is 2368+$|RID^*| + |t_3| + |RS|$ bits, where $|RID^*|$ denotes the binary length of RID^*.

We compare the communication overhead of P-SPAN and VSPN in result retrieval phase in Fig 5(b). We assume the length of navigation result RS in P-SPAN is equal to that in VSPN and $|RID^*| = |t_3| = 160$ bits. The communication overhead of the OBU is constant in our P-SPAN, while the overhead increases linearly with respect to the number of RSUs participating in navigation result generation in VSPN.

E. Storage Overhead

The storage overhead of OBUs is pretty low, as they need to maintain the public parameters $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, \hat{g}, \hat{g}_0, \mathcal{H})$

and the anonymous credential (AC, s) to access the parking navigation service. The public key certificates of the TA, cloud and RSUs can be sent along with the messages, such that it is unnecessary for OBUs to keep these certificates. To enable the retrievability of navigation results, the RSU maintains the Bloom filters CBF_K and VBF_K to hold the storage addresses of navigation results. The navigation results possess the fixed storage spaces determined by the length of navigation results and the number of results. CBF_K has the constant binary length. Support that an RSU can at most store n navigation results and CBF_K uses an array of m λ-bit counters to represent the number of collisions. CBF_K possesses λm bits on the RSU. The Bloom filter VBF_K maintains the shares of storage addresses. VBF_K is $m\gamma$ bits. In P-SPAN, if all locations in VBF_K are possessed when inserting a new address, the RSU has to build another VBF_K to store this address. Thus, the storage cost is doubled. To reduce the storage cost, we design the extended VBF_K in VI-B. In the extended VBF_K, if a collusion happens during the insertion of a new address, a new γ-bit string is added below $VB_{h_{k+\xi}(K)}$ if the location on $h_{k+\xi}(K)$ has not been occupied and the locations on $\{h_1(K), \cdots, h_{k+\xi-1}(K)\}$ are possessed, until $\xi = k'$. Therefore, if a collision happens, the length of the extended VBF_K will increase γ bits. Thereby, the length of the extended VBF_K is $(m+\eta)\gamma$ bits if η collisions happen in storage address insertion. Now we discuss the probability of η collisions in the extended VBF_K.

Having $(m, n, k, H) - VBF_K$, the false positive probability of a Bloom filter [34] is

$$F_{m,n,k} = \frac{m!}{m^{k(n+1)}} \sum_{i=1}^{m} \sum_{j=1}^{i} (-1)^{i-j} \frac{j^{kn} j^k}{(m-i)!j!(i-j)!}. \quad (10)$$

$F_{m,n,k}$ is the probability of at least one collision happens in the extended VBF_K. Thus, the probability that no collision occurs in the extended VBF_K is $\mathbb{P}_0 = 1 - F_{m,n,k}$. The probability of at least two collisions happens is $F_{m,n-1,k}F_{m,n,k}$, and thereby, the probability that exactly one collision occurs in the extended VBF_K is

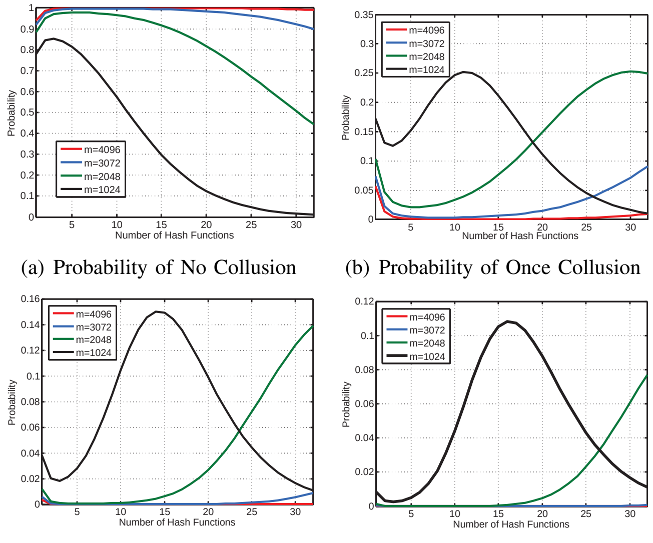
$$\mathbb{P}_1 = F_{m,n,k} - F_{m,n-1,k}F_{m,n,k}. \quad (11)$$

The probability that exactly η collisions happen in the extended VBF_K is

$$\mathbb{P}_\eta = \prod_{j=0}^{\eta-1} F_{m,n-j,k}(1 - F_{m,n-\eta,k}). \quad (12)$$

Therefore, the length of the extended VBF_K is $(m+\eta)\gamma$ bits with the probability of $\mathbb{P}_\eta$.

To clarify the collision probability of VBF_K, which impacts the storage overhead of the RSU, Fig. 6 illustrates the collusion probability in the extended VBF_K with $n = 256$, $m \in \{1024, 2048, 3072, 4096\}$ and $k \in \{1, 2, \cdots, 32\}$. Fig. 6(a) shows the probability that no collision happens with respect of the number of hash functions. With the increasing of m from 1024 to 4096, the collusion probability decreases significantly. The probability of no collision is largest if $m = 4096$, in which $\mathbb{P}(4096, 256, 11)$=0.99954, indicating that the probability of collisions is only 0.00046. Fig. 6(b), Fig. 6(c) and Fig. 6(d) depict the probability of collisions happening once, twice and thrice in the extended VBF_K, respectively. The collusion probability is fluctuant when $m = 1024$, 2048 and 3072, but it always larger than the collision probability when $m = 4096$ if k is less than 32. Therefore, if $m = 4096$ and $k = 11$, the collusion probability is the lowest in all settings as shown in Fig. 6. To maintain 256 navigation results, γ is at least 8 bits. If the RSU directly stores the index of K and the storage address, this index possesses 128 Kbytes, while the VBF_K is only 4Kbytes for the maintenance of storage addresses of navigation results.



(a) Probability of No Collusion

(b) Probability of Once Collusion

(c) Probability of Twice Collusion

(d) Probability of Thrice Collusion

Fig. 6. Probability of Collusion in the Extended VBF_K

VII. Conclusions

In this paper, we have proposed a privacy-preserving smart parking navigation system (P-SPAN) using Bloom filter and vehicular communications. In P-SPAN, a driver is allowed to query the available parking space to a cloud and retrieve the navigation result with privacy preservation. We have also developed an efficient data retrieving mechanism to enhance the retrieving probability of navigation results for anonymous vehicular communications under the fact that it is difficult for vehicles to hold the connections with the RSUs due to the high mobility. In addition, we have extended the Bloom filter to reduce storage overhead and collusion probability for RSUs. Finally, we have demonstrated that P-SPAN reaches the desirable security and privacy goals, and shown its efficiency and practicality for implementation in performance evaluation. For the future work, we will design a privacy-preserving roadside parking navigation system using crowdsourcing to achieve the roadside parking spaces discovery for drivers.

References

[1] J. Ni, K. Zhang, X. Lin, Y. Yu, and X. Shen, "Cloud-Based Privacy-Preserving Parking Navigation through Vehicular Communications," in *Proc. of SecureComm*, 2016, pp. 85–103.

[2] R. Lu, X. Lin, H. Zhu, and X. Shen, "An Intelligent Secure and Privacy-Preserving Parking Scheme through Vehicular Communications," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2772–2784, 2010.

[3] P. S. White, "No Vacancy: Park Slope's Parking Problem," http://www.transalt.org/news/releases, Feb. 27, 2007.

[4] H. Li, M. Dong, and K. Ota, "Control Plane Optimization in Software-Defined Vehicular Ad Hoc Networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7895–7904, 2016.

[5] R. G. Thompsona, K. Takadab, and S. Kobayakawa, "Optimisation of Parking Guidance and Information Systems Display Configurations," *Transp. Res. Pt. C- Emerg. Tech.*, vol. 9, no. 1, pp. 69–85, 2001.

[6] SWARCO, "Parking Guidance and Driver Information," http: www.ssspl.org/uploads/Products/Pdf/ParkingGuidancesystem.pdf, Aug. 02, 2002.

[7] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A Secure and Privacy-Preserving Protocol for Vehicular Communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442–3456, 2007.

[8] P. R. Pereira, A. Casaca, J. J. Rodrigues, V. N. Soares, J. Triay, and C. Cervelló-Pastor, "From Delay-Tolerant Networks to Vehicular Delay-Tolerant Networks," *IEEE Commun. Surv. Tutor.*, vol. 14, no. 4, pp. 1166–1182, 2012.

[9] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient Conditional Privacy Preservation Protocol for Secure Vehicular Communications," in *Proc. of IEEE INFOCOM*, 2008, pp. 1903–1911.

[10] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. Shen, "Security and Privacy for Smart City Applications: Challenges and Solutions," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 122–129, 2017.

[11] L. Guo, M. Dong, K. Ota, Q. Li, T. Ye, J. Wu, and J. Li, "A Secure Mechanism for Big Data Collection in Large Scale Internet of Vehicle," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 601–610, 2017.

[12] P. Vijayakumar, M. Azees, A. Kannan, and L. J. Deborah, "Dual Authentication and Key Management Techniques for Secure Data Transmission in Vehicular Ad Hoc Networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1015–1028, 2016.

[13] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An Identity-Based Security System for User Privacy in Vehicular Ad Hoc Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 9, pp. 1227–1239, 2010.

[14] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight Three-Factor Authentication and Key Agreement Protocol for Internet-Integrated Wireless Sensor Networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[15] J, Angwin and J. Valentinó-DeVries, "Apple, Google Collcet User Data," *The Wall Street Journal*, http://www.wsj.com/articles, Apr. 22, 2011.

[16] J. Cheng, "How Apple Tracks Your Location Without Consent, and Why It Matters," *Arstechnica*, http://arstechnica.com/apple/2011/04, Apr. 20, 2011.

[17] D. J. Wu, J. Zimmerman, J. Planul, and J. C. Mitchell, "Privacy-Preserving Shortest Path Computation," in *Proc. of NDSS*, 2016, pp. 1–41.

[18] T. Berman, F. Ferreira, and A. Valiente, "7 Things Car Thieves Know That You Do Not," *ABC News,* http://abcnews.go.com/Business/things-car-thieves/story?id=20938096, Nov. 20, 2013.

[19] M. D. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, "Location-Aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, 1590–1602, 2007.

[20] J, Huang, L. Yeh, and H. Chien, "ABAKA: An Anonymous Batch Authenticated and Key Agreement Scheme for Value-Added Services in Vehicular Ad Hoc Networks," *IEEE Trans. Veh. Technol.* vol. 60, no. 1, pp. 248–262, 2011.

[21] T. Chim, S. Yiu, L. C. Hui, and V. O. Li, "VSPN: Vanet-Based Secure and Privacy-Preserving Navigation," *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 510–524, 2014.

[22] W. Cho, Y. Park, C. Sur, and K. H. Rhee, "An Improved Privacy-Preserving Navigation Protocol in VANETs," *J. Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 4, no. 4, pp. 80–92, 2013.

[23] C. Sur, Y. Park, and K. H. Rhee, "An Efficient and Secure Navigation Protocol based on Vehicular Cloud," *Int. J. Comput. Math*, vol. 93, no. 2, pp. 325–344, 2016.

[24] D. Pointcheval and O. Sanders, "Short Randomizable Signatures," in *Proc. of CT-RSA*, 2016, pp. 111–126, 2016.

[25] D. Boneh and X. Boyen, "Short Signatures without Random Oracles," in *Proc. of EUROCRYPT*, 2004, pp. 56–73.

[26] M. Bellare and O. Goldreich, "On Defining Proofs of Knowledge," in *Proc. of CRYPTO*, 1992, pp. 390–420.

[27] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract)," in *Proc. of STOC*, 1985, pp. 291–304.

[28] A. Fiat and A. Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," in *Proc. of CRYPTO*, 1986, pp. 186–194.

[29] C. Dong, L. Chen, and Z. Wen, "When Private Set Intersection Meets Big Data: An Efficient and Scalable Protocol," in *Proc. of ACM CCS*, 2013, pp. 789–800.

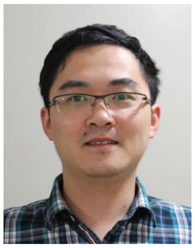
[30] M. H. Au, J. K. Liu, J. Fang, Z. L. Jiang, W. Susilo, and J. Zhou, "A New Payment System for Enhancing Location Privacy of Electric Vehicles," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 3–17, 2014.

[31] W. Xu, H. A. Omar, W. Zhuang, and X. Shen, "Delay Analysis of In-Vehicle Internet Access via On-Road WiFi Access Points," *IEEE Access*, vol. 5, no. 1, pp. 2736–2746, 2017.

[32] S. Skiena, "Multiprecision Integer and Rational Arithmetic C/C++ Library," http://www. freshports.org/math/miracl/, Jul. 10, 2008.

[33] J. L. Beuchat, J. E. González-Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T. Teruya, "High-Speed Software Implementation of the Optimal Ate Pairing over Barreto-Naehrig Curves," in *Proc. of Pairing*, pp. 21–39, 2010.

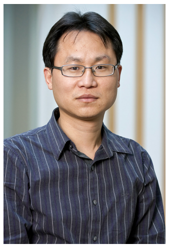
[34] K. Christensen, A. Roginsky, and M. Jimeno, "A New Analysis of The False Positive Rate of a Bloom Filter," *Inf. Process. Lett.*, vol. 110, no. 21, pp. 944–949, 2010.



Jianbing Ni (S'16) received the B.E. degree and the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2011 and 2014, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests are applied cryptography and information security, with current focus on cloud computing, smart grid, mobile crowdsensing, fog computing and Internet of Things.

Kuan Zhang (S'13-M'17) received the B.Sc. degree in Communication Engineering and the M.Sc. degree in Computer Applied Technology from Northeastern University, China, in 2009 and 2011, respectively. He received the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Canada, in 2016. He was also a postdoctoral fellow with the Broadband Communications Research (BBCR) group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. Since 2017, he has been an assistant professor at the Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, USA. His research interests include security and privacy for mobile social networks, e-healthcare systems, cloud computing and cyber physical systems.

Yong Yu (M'16) is currently a Professor of Shaanxi Normal University, China. He holds the prestigious one hundred talent Professorship of Shaanxi Province as well. He received his Ph.D. degree in cryptography from Xidian University in 2008. He has authored over 60 referred journal and conference papers. His research interests are cryptography and its applications, especially public encryption, digital signature, and secure cloud computing. He is an Associate Editor of Soft Computing.



Xiaodong Lin (M'09-SM'12-F'17) received the Ph.D. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, and the Ph.D. degree in electrical and computer engineering (with Outstanding Achievement in Graduate Studies Award) from the University of Waterloo, Waterloo, ON, Canada. He was an Associate Professor of information security with the Faculty of Business and Information Technology, University of Ontario Institute of Technology (UOIT), Canada. He is currently an Associate Professor with the Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, ON, Canada. His research interests include wireless network security, applied cryptography, computer forensics, software security, and wireless networking and mobile computing.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. Dr. Shen is a University Professor, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, Infocom'14, IEEE VTC'10 Fall, and Globecom'07, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08, the General Co-Chair for ACM Mobihoc'15, Chinacom'07 and QShine'06, the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for IEEE Internet of Things Journal, IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo, the Joseph LoCicero Award and the Education Award 2017 from the IEEE Communications Society. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.