Privacy-preserving Partner Selection for Ride-sharing Services

Yuanyuan He, *Student Member, IEEE,* Jianbing Ni, *Student Member, IEEE,* Xinyu Wang, Ben Niu, *Member, IEEE,* Fenghua Li, *Member, IEEE,* Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—Ride-Sharing Services (RSSs) assist drivers to find proper riders for vacant seats on the road, providing appealing benefits of shared travel cost and improved vehicle occupancy, which have revolutionized transportation business, as witnessed by the success of Lyft Line, UberPool and Waze Carpool. Selecting proper ride-share partners for drivers based on riders' trip data is essential for RSSs, but it also leads to the exposure of drivers' and riders' future locations and trajectories. To preserve the individual privacy during partner selection, in this paper, we propose a privacy-preserving ride matching scheme for selecting feasible ride-share partners in RSSs. Firstly, we design a spatial region-based selection mechanism, which allows the ride sharing-server (RS-server) to pre-choose riders in the matched regions with drivers, without exposing their accurate sources and destinations. Secondly, with the encrypted itineraries of drivers and riders, the RS-server further selects potential ride-share partners according to the Travel Time Saving (TTS) and the feasibility of time schedules. Thirdly, the RS-server determines proper ride-share partners with the objective of maximizing the system-wide TTS. With the three-step partner selection, suitable riders can be discovered for the drivers to share vacant seats, resulting in the saving of total travel time and expenditure for riders and drivers. Finally, we demonstrate that the proposed scheme offers strong privacy guarantees to both riders and drivers while maintaining the efficiency and practicality of RSSs.**

I. Introduction

Ride-Sharing Services (RSSs) provide partner discovery services to drivers and riders with similar rides for initializing sharing travel experiences. RSSs allow drivers to share vacant seats of their vehicles on the road, bringing about various benefits to individual users (e.g., improved vehicle occupancy, shared travel costs and extended social circles) and the society (e.g., reduced traffic congestion, fuel consumption and carbon dioxide emissions) [1], [2]. Due to these appealing advantages, many service providers have emerged to offer ride-share partner discovery services, e.g. Flinc, Lyft Line, UberPool, Waze Carpool and Blablacar. Ride-sharing has become increasing popular in metropolis to reduce crowded traffic and expensive transportation costs.

Yuanyuan He, Xinyu Wang, Ben Niu and Fenghua Li are with State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China. Email: {heyuanyuan, wangxinyu, niuben, lfh}@iie.ac.cn.

Yuanyuan He, Xinyu Wang and Fenghua Li are with School of Cyber Security, University of Chinese Academy of Sciences, China.

Yuanyuan He, 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.

Ben Niu is the Corresponding Author.

To initialize a sharing trip, a driver sends a ride offer or a rider sends a ride request to a Ride-Sharing server (RS-server) for finding rider-share partners with similar itineraries. As shown in Fig. 1, Alice plans to drive from London to Manchester during 8:00-16:00 on June 3 and delivers the information to the RS-server to find a rider. The RS server, acting as the service provider of a RSS, discovers a matched rider (i.e., Bob) for Alice based on their trip data. If both Alice and Bob agree to share the trip, the RS-server facilitates the communication between them and usually charges a commission for the successful ride-share. Alice and Bob can start the trip and share fuel costs.

Despite the appealing advantages of RSSs, this popular travel mode meanwhile brings about various challenges, one of which is the privacy leakage [3], [4]. To initialize a ride-share, both drivers and riders are required to deliver their maintained itineraries, including sources, destinations, routes and time schedules, to the RS-server for partner selection. However, the RS-server may not be fully trusted, it has numerous motivations to share the detailed trip data with their cooperators for monetary reasons. For example, to promote the new platform, Uber Movement, Uber has released staggering 2 billion pieces of trip data collected from people in more than 450 cities [5]. Further, data exposure accidents frequently happened on the data centers of enterprises, such as Yahoo [6] and Apple [7], dramatically reduced users' trust in application providers. Thus, it is difficult to believe that all the trip data and historical rides from millions of drivers and riders would not be revealed to the public. From these data, it is possible for attackers to acquire the future locations and trajectories of a specific rider, such as a profile politician, celebrity and even personal acquaintances [8]. They further become targets of blackmail, rob or sexually attacks [3], [9]. For example, an attacker can easily estimate how long the target rider is away from home and use this information to plan a burglary, and the curious RS-server is able to predict the commuting custom and living styles to promote unwelcome advertisements. Therefore, it is of importance to protect the privacy of drivers and riders against the RS-server while maintaining the beneficial features of RSSs.

Currently, both industry and academia have introduced several approaches to protect individual users' privacy during ride-share partner selection. To achieve this, many schemes hide a user's identity with pseudonyms [10], [11], [12], [13], but highly subtle information about the user's home address, job, social statues, healthy status, personal preference and date of birth, can be inferred by analyzing location data integrated

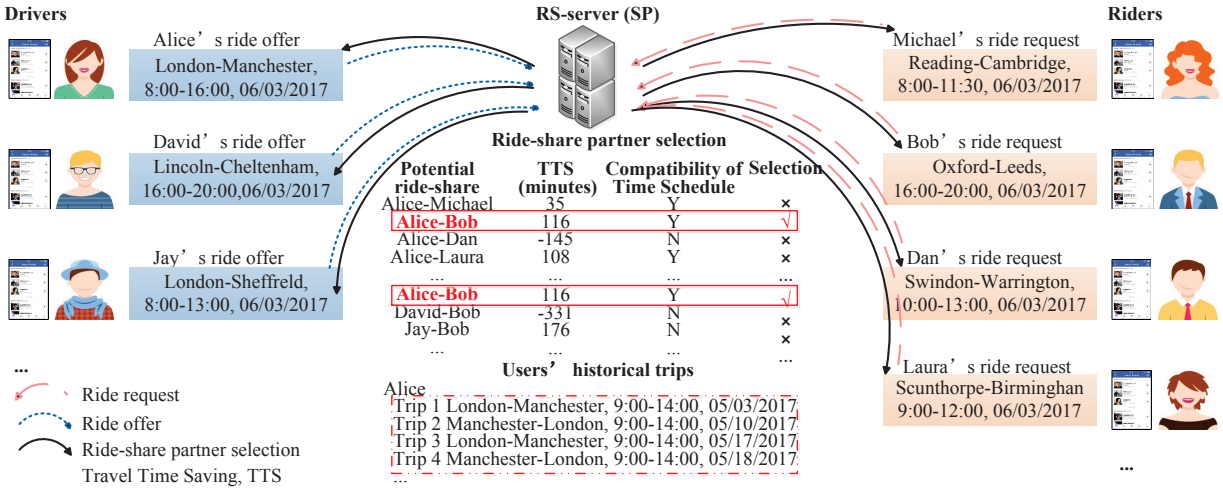


Fig. 1. RSSs

with other knowledge, such as social graph and cellular data [14], [15]. To cut the privacy leakage, several location privacy-enhancing solutions for ride sharing and ride hailing have been proposed based on location obfuscation techniques or encryption tools in the privacy computing category. The location obfuscation techniques such as spatial cloaking and k-anonymity [16] are highly efficient to protect riders' locations, but they offer limited privacy preservation due to the trade-off between the accuracy of ride matching results and the size of a cloaked area [17]. Encryption techniques, such as Searchable Symmetric Encryption (SSE) [18], Private Set Intersection (PSI) [19] and Somewhat Homomorphic Encryption (SHE) [20], provide relatively high level of privacy preservation at the cost of computational efficiency. Approaches based on these cryptographic techniques only support simple location proximity measurements for partner selection, such as Manhanttan distance or Euclidean distance, which lead to the improper rider matching and thereby undesired waste on travel time and fuels [21]. Map distance is a better factor to measure the driving time between two points on the road. It illustrates the precise time cost for a driver to travel from one point to another. The travel time saving (TSS) brought by sharing multiple trips has become the main motivation for users to participate in RSSs. TTS should be one of the essential conditions to determine the feasibility of a ride-share match in RSSs, apart from the matching of sources, destinations, departures and arrive times for both drivers and riders.

In this paper, we propose a novel Privacy-preserving RIde-share partner Selection scheme (PRIS) to resolve users' privacy leakage problem and achieve map distance-based partner selection in RSSs, simultaneously. By considering the feasibility of ride-share match and system-wide TTS, we design a three-step mechanism to determine ride-share partners over encrypted trip data for each driver without sacrificing the effectiveness and efficiency in RSSs. The major contributions of our paper are four-fold.

- The PRIS is proposed to address the conflict between privacy leakage and partner selection in RSSs. With the designed three-step partner selection, including spatial region-based pre-selection, feasible TTS-based ride matching and maximal system-wide TTS optimization, we can significantly reduce the total travel time cost for both drivers and riders, and cut the threats on their privacy leakage.
- In spatial region-based pre-selection, the ride requests are preselected based on drivers' spatial regions without exposing their sources and destinations. By testing the equality of locations encrypted in ride offers and requests, the RS-server preselects riders such that their pickup and drop-off points are within the driver's acceptable spatial region.
- To find the proper ride-share match, the ride requests are further filtered based on the positive TTS calculated from map distance, ensuring the ride-share match can save time and fuels. The RS-server utilizes Paillier cryptosystem to calculate the TTS and test the feasibility of ride-share match without learning the concrete itineraries.
- The proper ride-share partners are finally determined based on the system-wide TSS. By solving the weight bipartite optimization problem, the RS-server can determine the ride-share partners while achieving maximized total TTS.

The remainder of the paper is organized as follows. We review the related works in Section II, and present the system model, security model and design goals in Section III. Then, we propose the PRIS in Section III-A, followed by the security discussion in Section V. Finally, we evaluate the performance of the proposed scheme in Section VI, and draw the conclusion in Section VII.

II. Related Works

The existing privacy-preserving partner selection solutions for RSSs (e.g., car pooling and ride hailing) are based on either non-encryption or encryption mechanisms. Friginal et al. [22] introduced the problem of privacy leakage in carpooling system and proposed a privacy-preserving dynamic carpooling framework following the privacy-by-design principle. Goel *et al.* [14] proposed a privacy-aware ride matching scheme based on the obfuscation technique, in which the degree of privacy preservation depends on the degree of imprecision brought by the riders' obfuscated locations. Some imprecision-based approaches have been designed [17], [23] to ensure riders'

privacy in ride-share partner selection. The representative work is PrivateRide [17], which uses the spatial cloaking technique to protect riders' location. PrivateRide achieves high efficiency, yet its ride matching result is imprecise and it cannot guarantee the same level of privacy to all riders. In fact, the level of privacy preservation is affected by the trade-off between the size of a cloaked area and the accuracy of ride matching result. Other non-encryption-based solutions, such as cloaking [24], mix zone [25], anonymity [26] and geographic masking [27] are leveraged to address the problem of privacy leakage during partner selection in RSSs, but most of these schemes utilize the expanded geographic areas to hide the exact region of riders for privacy preservation, resulting in the improper matching of trips for both drivers and riders. To promote privacy preservation and ease improper ride matching result, some encryption-based schemes have been proposed in RSSs. Xi *et al.* [28] integrated the Private Information Retrieval (PIR) and additive homomorphic encryption techniques to privacy-preserving shortest path computation. Nonetheless, computational overhead is quite high in [28]. To improve the efficiency of PIR-based privacy-preserving shortest path computation, Wu *et al*. [29] have applied a graph compression algorithm on road networks. Based on the privacy-preserving shortest path computation approaches, some schemes [23], [30] further consider the privacy-preserving meeting point computation for ride-share partner selection to reduce the driver's detour and improve the number of successful trip matching. The privacy-preserving partner selection and meeting point computation for RSSs are actually based on secure multi-party computation of proximity testing [31] in the privacy computing [32] category. Bilogrevic *et al.* [30] integrated PSI techniques and multimodal shortest path algorithm to achieve secure multi-party computation of location proximity testing for privacy-preserving optimal meeting point calculation. The naive location proximity is widely used in privacy-preserving partner selection [17], [21], [33], such as Euclidean distance, Manhattan distance, etc. Unfortunately, it is not precise enough to meet users' personal requirements for ride-sharing, thus some comprehensive selection rules are build based on the location proximity, which considering many other equally significant factors, such as TTS, travel cost saving [1], reputation rating [34] and information from social network [35]. Some privacy-enhancing approaches [21], [33] incorporate the comprehensive selection rule and SHE to improve user experience in aspects of personal preference, benefit and trust, while achieving strong privacy preservation. However, most of them suffer from the weakness of imprecise ride-share matching based on Euclidean distance or Manhattan distance [21], due to the limited operations supported by two-party or multi-party secure computation [33], [35]. Compared to Euclidean distance and Manhattan distance, map distance is relatively precise to measure location proximity between users. The travel time based on map distance would help test the feasibility of a ride-share match and calculate potential TTS of the joint trip with relatively high accuracy [1], which are the primary selection rules in RRS and have not been mentioned in the most of existing privacy-preserving solutions. Hence it is necessary to propose a privacy-enhanced scheme supporting map distance-based selection rules.

Therefore, we study the feasibility of ride-share matches and the TTS calculation of corresponding shared trip considering map distance, and propose a privacy-preserving scheme to achieve ride-share partner selection with the protection of both drivers' and riders' trip data against the RS-servers and other individual users based on data encryption techniques. A three-step partner selection mechanism is designed to reduce the total travel time cost for both drivers and riders, and cut the threats on their privacy leakage.

III. Problem Statement

In this section, we present the system model, security model, and identify our design goals for ride-share partner selection.

A. System Model

The ride-sharing system consists of a large number of users and a RS-server.

• *Users*: A user is either driver $D_i \in U_D = \{D_1, D_2, \cdots, D_n\}$ or rider $R_s \in U_R = \{R_1, R_2, \cdots, R_n\}$ with trip tr_i or tr_s. The users use their mobile phones with ride-sharing applications to access PRIS provided by the RS-server. Rider R_s has to locally specify her/his trip tr_s, including source v_s, destination w_s, the earliest departure time t_{v_s} and the latest arrive time t_{w_s}, and then encrypts it to generate ride request Rtr_s, as well as driver D_i. Additionally, to form a ride offer, driver D_i needs to set and encrypt her/his acceptable spatial region set Q_i, including all locations where the driver is willing to pick up or drop off a rider in her/his trip. All ride offers and ride requests are sent to the RS-server. With PRIS, they obtain the information about the partners recommended by the RS-server. The driver and rider can start their travel, if both accept this recommendation.

• *RS-server*: A RS-server is a platform with computational and storage capabilities offering the RSS for particular metropolitans. The RS-server selects feasible matches between ride offers $\{Dtr_i\}_{D_i \in U_D}$ and ride requests $\{Rtr_s\}_{R_s \in U_R}$. The RS-server further selects the best ride-share partner R_i^*/D_s^* for user D_i/R_s and arranges ride-share with objectives in accordance to benefits, e.g., maximizing system-wide TTS, and returns results to the riders and drivers.

The system model is shown in Fig. 2. To access RSSs, driver D_i or rider R_s locally generates a ride offer/request, i.e. Dtr_i/Rtr_s. Take driver Alice for example, Alice sets the trip plan $tr_A = (Alice, v_A = London, w_A = Manchester, t_{v_A} = 08:00, t_{w_A} = 16:00, Q_A)$, where Q_A is a spatial region supporting taking detours. She calculates the hash values of locations $\{h(l_k)\}_{l_k \in Q_A}$ and evaluates the vehicle time values from v_A to w_A, from l_k to w_A and from v_A to l_k, i.e. $t_{v_A,w_A}, t_{l_k,w_A}, t_{v_A,l_k}$. Then Alice encrypts these information with her secret integer a_i and public key pk_i to form ride offer $Dtr_A = \{driver\ Alice, \{(h(l_k))^{a_i}, \sigma'_{v_A,l_k}, \sigma'_{w_A,l_k}\}_{l_k \in Q_A}, h_0^{a_A}, \sigma_{v_A,w_A}, \sigma'_{v_A}, \sigma_{w_A}\}$. Meanwhile, rider R_s generates ride request $Rtr_s = \{rider\ R_s, h_0^{b_s}, (h(v_s))^{b_s}, (h(w_s))^{b_s}\}$, where h_0, b_s, v_s and w_s are the public parameter, the rider's secret integer, source and destination, respectively.

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.2018.2809039, IEEE Transactions on Vehicular Technology

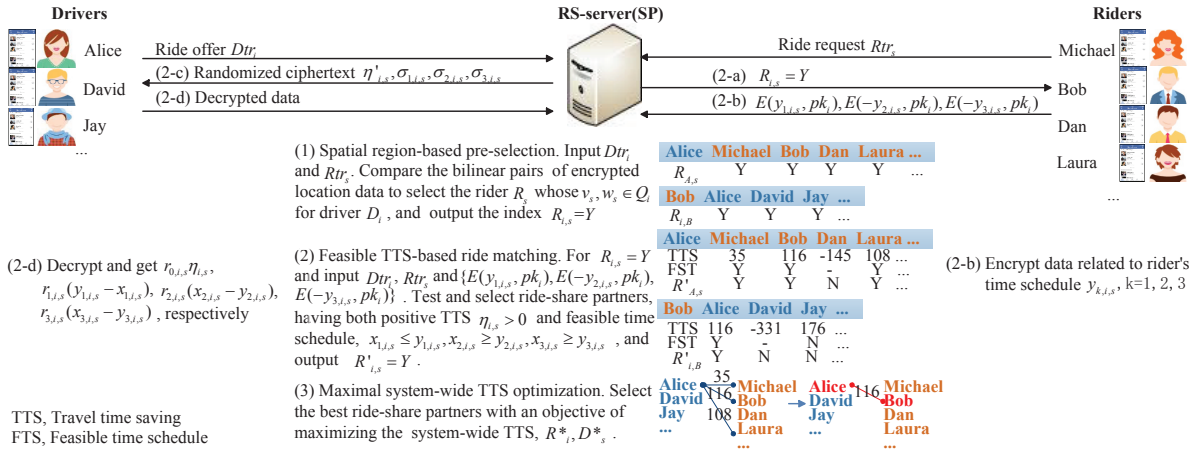


Fig. 2. System model

After receiving the encrypted ride offers/requests from users, the RS-server uses the three-step partner selection mechanism to discover feasible partners for ride-sharing.

Firstly, the RS-server preselects riders for any driver D_i over encrypted data in ride offer Dtr_i and ride requests $\{Rtr_s\}_{R_s \in U_R}$ (step 1). The pre-selection rule is that a rider would be preselected for a driver if the driver is willing to take a detour to accommodate the rider, meaning that the preselected rider's v_s and w_s should be within the driver's pre-defined acceptable spatial region set Q_i. For Alice, the RS-server calculates the bilinear pairs of the ciphetexts $\{(h(l_k))^{a_i}\}_{l_k \in Q_A}$ in ride offer Dtr_A and $\{h_0^{b_s}\}_{R_s \in U_R}$, and that of $h_0^{a_A}$ and $((h(v_s))^{b_s}, (h(w_s))^{b_s})$ in ride requests $\{Rtr_i\}_{R_s \in U_R}$ to preselect riders such that riders' sources and destinations are within Q_A and sets $R_{A,s} = Y$. As shown in Fig. 2, riders $\{Michael, Bob, Dan, Laura, \cdots\}$ are preselected for Alice. For the preselected rider R_s, the RS-server can pick out ciphertext $(\sigma'_{v_A,v_s}, \sigma'_{w_A,w_s})$ from $\{\sigma'_{v_A,l_k}, \sigma'_{w_A,l_k}\}_{l_k \in Q_A}$ in Alice's ride offer to prepare for TTS calculation. Similarly, drivers $\{Alice, David, Jay, \cdots\}$ are preselected for Bob.

Secondly, the RS-server discovers feasible ride-share matches from preselected pairs without knowing individual users' trip data (step 2). The feasible ride-share match is required to have both positive TTS and feasible time schedule. The TTS between Alice and rider R_s is denoted by $\eta_{A,s} = (t_{v_A,w_A} + t_{v_s,w_s}) - (t_{v_A,v_s} + t_{v_s,w_s} + t_{w_s,w_A})$. The positive $\eta_{A,s}$ means that the travel time of their joint trip is shorter than the sum of separate travel time values, so Alice and rider R_s have the incentive to share their rides. The feasible time schedule refers to no confliction of Alice's and the rider's time ranges. The RS-server communicates with Alice and each preselected rider R_s and executes operations on ciphertexts $\{(\sigma'_{v_A,v_s}, \sigma'_{w_A,w_s}), R_s \in U_R, \sigma_{v_A,w_A}, \sigma'_{v_A}, \sigma_{w_A} \in Dtr_A\}$ to calculate corresponding TTS value $\eta_{A,s}$ and test whether the potential shared trip is time feasible or not. The feasible ride-share match is set as $R'_{A,s} = Y$. Riders {Michael, Bob, Laura, $\cdots$} are feasible ride-share parters for Alice, while rider Dan is filtered due to the negative TTS (-145 minutes) and the infeasible time schedule (10:00-13:00). Similarly, drivers $\{David, Jay, \cdots\}$ meanwhile fail to share their rides with rider Bob due to negative TTS or infeasible time schedule.

Thirdly, the RS-server selects ride-share partners for all drivers with the purpose of maximal system-wide TTS (step 3). The RS-server constructs a weight bipartite based on all feasible ride-share matches ($R'_{i,s} = Y$). For an instance, an edge between Alice and rider R_s represents a feasible ride-share between them, i.e. $R'_{A,s} = Y$, and the weight represents corresponding TTS value $\eta_{A,s}$. The RS-server views the optimal ride-share partner selection problem on the weight bipartite as a maximal weight bipartite matching optimization problem and solves it to select the best ride-share partner (R^*_i, D^*_s) for every user (D_i, R_s).

With PRIS, the matched driver and rider share their rides with a low total time cost. Alice can share her trip with Bob, since their shared trip is feasible and brings a large TTS, 116 minutes. If the RS-server needs to select multiple riders for a driver, steps 1, 2 and 3 can be repeated multiple times.

B. Security Model

In general, most of drivers and riders are honest to access RSSs and enjoy the benefits, but a few users may be curious about the personal information about others, such as their friends' locations. Some may eavesdrop on the communication channels to capture their interested data. For example, a user may try to obtain the riders' information to build a ride-sharing trip by himself without payment to the RS-server. Further, it is impossible to fully believe the service provider of RSS. The RS-server is assumed to follow the agreements made with users, such as privacy policy, in which the RS-server claimed to keep all the users' trip data private. However, it is widely believed that the service provider may break the privacy policy for its own purpose. Besides, the RS-server may not actively collude with a user, since this behavior impacts its reputation and gives others the witness of its misbehavior. We model the misbehavior of the RS-server and users as follows:

- **AS1**: The RS-server maintains information about riders and drivers (e.g. personal information and trip data) in a long period to do large-scale inference attacks [36] for profiling riders' and drivers' activities.
- **AS2**: The RS-server might attempt to learn more information about specific riders, such as the trip data [21],

[37]. The RS-server knows the precise pick-up location and time of a specific rider, and wants to know the drop-off location and time of the rider, or vice versa.

- **AU**: The drivers/riders might attempt to capture some sensitive information about specific riders/drivers, including personal information and travel data. The drivers/riders input fake trip data frequently and want to know the target rider's/driver's source, destination, departure and arrive time.

C. Design Goal

Our design goal is to achieve ride-share partner selection without disclosing any sensitive information about users in RSSs. Specifically, our goals can be divided into three folds:

1) Functions: To achieve ride-share partner selection, the RS-server should support feasible ride-sharing and optimized arrangement with the objective of maximal system-wide TTS.
2) Privacy Preservation: The starting/ending location and time range of each trip should be preserved to prevent privacy leakage from RS-sever (**AS1**, **AS2**) and privacy exposure to users (**AU**).
3) Efficiency: The computational cost and communication overhead should be low enough to ensure the normal functions of smart phones.

IV. Proposed Scheme

In this section, we review the preliminaries and present the PRIS scheme for RSSs.

A. Preliminaries

Paillier cryptosystem [38] used in PRIS is an additive homomorphic encryption. The details are presented as follows:

1) Key Generation: $(pk, sk) \leftarrow KeyGen(\kappa)$. An entity chooses two primes p_0, p_1 and computes $N = p_0 p_1$ and $\lambda = lcm(p_0 - 1, p_1 - 1)$. Then, she/he selects a random $g \in Z^*_{N^2}$ such that $gcd(L(g^\lambda \bmod N^2), N) = 1$, where $L(x) = \frac{(x-1)}{N}$. The entity's Paillier public key and private key are $pk = (N, g)$ and $sk = \lambda$, respectively.
2) Encryption: $c \leftarrow E(m, pk)$. Let $m \in Z_N$ be a plaintext and $r \in Z_N$ be a random number. The ciphertext is given by $c = E(m \bmod N, r \bmod N) = g^m r^N \bmod N^2$.
3) Decryption: $m \leftarrow D(c, sk)$. Given a ciphertext $c \in Z_{N^2}$, the corresponding plaintext can be derived as $m = \frac{L(c^\lambda \bmod N^2)}{L(g^\lambda \bmod N^2)} \bmod N$.

The Paillier cryptosystem has two important properties:

1) Homomorphism. For any $m_1, m_2, r_1, r_2 \in Z_N$, we have
$$E(m_1, r_1)E(m_2, r_2) = E(m_1 + m_2, r_1 r_2) \bmod N^2, \quad (1)$$
$$E^{m_2}(m_1, r_1) = E(m_1 m_2, r_1^{m_2}) \bmod N^2. \quad (2)$$
2) Self-blinding. $E(m_1, r_1) r_2^N = E(m_1, r_1 r_2) \bmod N^2$ which implies that any ciphertext can be randomized without learning the plaintext.

TABLE I
Syntax

Syntax	Descriptions
RSSs	Ride-Sharing Services
RS-server	Ride-Sharing server
TTS	Travel Time saving
AS1, AS2	The two kinds of attack launched by the RS-server
AU	The attack launched by the drivers or riders
PRIS	Privacy-preserving RIde-share partner Selection scheme
ORide	Oblivious Ride [1]
EP	EndPoint-based matching scheme [2]

TABLE II
Notations

Notations	Descriptions
$D_i \in U_D$	Driver D_i, the driver set U_D
$R_s \in U_R$	Rider R_s, the rider set U_R
P	A public set P of locations in service region
$tr_i(tr_s)$	D_i's (R_s's) trip plan $tr_i = (D_i, v_i, w_i, t_{v_i}, t_{w_i}, t_{v_i,w_i}, Q_i)$, $tr_s = (R_s, v_s, w_s, t_{v_s}, t_{w_s}, t_{v_s,w_s})$
$(v_i, w_i), (v_s, w_s)$	The source and destination of D_i and that of R_s
$t_{v_i}, t_{w_i}, t_{v_i,w_i}$	The earliest departure time, the latest arrive time and the
$t_{v_s}, t_{w_s}, t_{v_s,w_s}$	expected travel time of D_i and that of R_s
$l_k \in Q_i$	A spatial region in which D_i is willing to take a detour
$t_{a,b}$	The driving time from location a to b
$\mathbb{G}$, $\mathbb{G}_T$	Two cyclic multiplicative groups of order q
p_0, p_1, q	Large primes
$Z^*_{N^2}$	A cyclic multiplicative group of order N^2, $N = p_0 p_1$
h_0	A public integer $h_0 \in \mathbb{G}$
e	A bilinear map $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$
(pk_i, sk_i)	Driver D_i's key pair of the Paillier cryptosystem
$a_i(b_s) \in Z^*_q$	Driver D_i's (rider R_s's) secret random integer
$h: P \rightarrow \mathbb{G}$	Hash function mapping locations into the cyclic group
$Dtr_i(Rtr_s)$	Driver D_i's (rider R_s's) ride offer (ride request)
$G(i, s) = \eta_{i,s}$	TTS generated by the ride-share match of D_i and R_s
$r_{k,i,s} \in Z_N$	Secret random integers chosen by the RS-server
$x_{j,i,s}, y_{j,i,s}$	The variables for testing feasibility of time schedules
$\eta'_{i,s}$	TTS value $\eta_{i,s}$ is encrypted with pk_i
$\{\sigma_{j,i,s}\}_{j=1}^3$	The ciphertexts correspond to variables $(x_{j,i,s}, y_{j,i,s})$
$R_{i,s}$	The result of spatial region-based pre-selection selection
$R'_{i,s}$	The result of feasible TTS-based ride matching
$x_{i,s} = 0, 1$	Whether D_i and R_s share a ride

B. Overview of PRIS

The PRIS consists of three components: initialization, ride offer/request generation and three-step ride selection.

In initialization, the RS-server bootstraps PRIS by defining the format of trip plan (tr_i for driver D_i and tr_s for rider R_s) and the feasible selection rules. The feasibility of ride-sharing includes two rules: the positive TTS and the feasibility of separate time schedules. The TTS is used to measure the proximity between the driver's and rider's trips. For a given pair of a driver and a rider, the problem of selecting a feasible ride-share partner is transformed into the problem of calculating TTS securely and the problem of testing whether Equ. (4), (6), (7) and (8) hold or not over encrypted data.

To share or acquire the vacant seats on vehicles, driver D_i or rider R_s encrypts its trip data in tr_i or tr_s to generate ride offer Dtr_i and ride request Rtr_s, respectively.

Based on all received ride offers and requests, the RS-server starts its three-step ride selection to select feasible ride-share partners with privacy preservation.

1) The RS-server calculates bilinear pairs of modular exponentiations in the $\mathbb{G}$, including $e((h(l_k))^{a_i}, (h_0)^{b_s}), l_k \in Q_i$, $e((h(v_s))^{b_s}, h_0^{a_i})$ and $e((h(w_s))^{b_s}, h_0^{a_i})$ for the given pair of Dtr_i and Rtr_s. a_i and $b_s \in Z^*_q$ are secret random numbers chosen by driver D_i and rider R_s separately. h_0 is the public random number chosen by the RS-server. Based on the bilinear pairs, the RS-server preselects riders who are within driver D_i's acceptable spatial region with privacy preservation. The RS-server sets $R_{i,s} =$ "Y" for the preselected pairs such that

$v_s, w_s \in Q_i$. Otherwise, $R_{i,s} =$ "N".

2) The RS-server interacts with drivers and riders having $R_{i,s} =$ "Y" to further select feasible ride-share matches by calculating TTS and testing inequalities Equ. (4), (6), (7) and (8). The privacy preservation relies on the additively homomorphic cryptosystem, Paillier cryptosystem. RS-server sets $R'_{i,s} =$ "Y" for the selected pairs. Otherwise, $R'_{i,s} =$ "N".

3) The RS-server selects the best ride-share partners on the purpose to maximize system-wide TTS ($max \sum_{i,s} x_{i,s} G(i,s)$) by solving maximal weight bipartite matching optimization problem. After finding the partner, the participants start their ride-sharing trip with reduced total time cost.

Table I and Table II provide the syntax and notations mentioned in PRIS.

C. The Detailed PRIS

We show the detailed description of PRIS.

① Initialization.

The RS-server sets up a bilinear group $(e, q, \mathbb{G}, \mathbb{G}_T)$ with a bilinear map: $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic multiplicative groups of a large prime order q. The RS-server sets a collusion-resistant hash function $h : \{0,1\} \rightarrow \mathbb{G}$ that maps a location l into an integer $h(l)$ in the $\mathbb{G}$. The RS-server chooses a public integer $h_0 \in \mathbb{G}$ randomly and broadcasts it to each legitimate user. Driver D_i and rider R_s choose secret random integers $a_i \in Z_q^*$ and $b_s \in Z_q^*$ separately to prepare for the generation of ride offer Dtr_i and ride request Rtr_i. Driver D_i generates a public-secret key pair of Paillier cryptosystem $(pk_i, sk_i) \leftarrow KeyGen(\kappa)$.

Since each individual user's route should be private, the user has incentives to plan trip route and evaluate travel time, and then sets her/his time range locally. It is convenient for every entity to evaluate the travel time between any two locations in the RS-server's service area P from some free navigation service providers (e.g. Google Map). Since both the route and map distance between a given pair of locations $v, w \in P$ are implied in the travel time from location v to w, any user can specify the travel time cost, the departure and arrive time instead of route and map distance. Rider R_s is required to specify the detailed information of her/his trip locally, consisting of a source v_s, a destination w_s, the earliest departure time t_{v_s} and the latest arrive time t_{w_s} that implies the rider's expected trip route. A reasonable time range $[t_{v_s}, t_{w_s}]$ should be longer than the rider's expected trip time cost t_{v_s,w_s},

$$t_{w_s} - t_{v_s} \geq t_{v_s,w_s}, \tag{3}$$

as well as driver D_i. Besides, driver D_i needs to set one more set Q_i that includes all locations where the driver is willing to take a detour to pick up or drop off one rider. The trip plan is defined as follows.

Definition 1. ***Trip Plan***. *Let $TR = \{tr_1, tr_2, \cdots\}$ be a set of trip plans, where a trip plan can be $tr_i = (D_i, v_i, w_i, t_{v_i}, t_{w_i}, t_{v_i,w_i}, Q_i)$ from a driver or $tr_s = (R_s, v_s, w_s, t_{v_s}, t_{w_s}, t_{v_s,w_s})$ from a rider. $D_i(R_s)$ represents the driver's (rider's) name. $\{v_i, w_i\}(\{v_s, w_s\})$ means the source and destination of driver D_i (rider R_s). t_{v_i}, t_{w_i} and t_{v_i,w_i} ($t_{v_s}, t_{w_s}, t_{v_s,w_s}$) are the driver D_i's (rider R_s's) earliest departure time, latest arrive time and expected travel time from her/his source to destination. The set $Q_i \subset P$ is a spatial region in which D_i is willing to take a detour.*

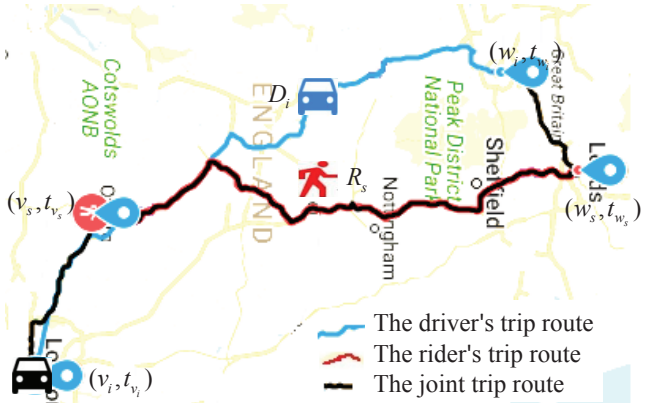


Fig. 3. The detour to accommodate a rider

There is no Q_s in rider R_s's trip plan, since the rider does not need to consider a detour with the assumption that a driver would share a ride with at most one rider.

The RS-server defines selection rules for ride matching. Assuming that driver D_i plans to travel from v_i to w_i within time range $[t_{v_i}, t_{w_i}]$ and is willing to make a detour to accommodate someone within a spatial region Q_i. Rider R_s is going to travel from v_s to w_s within time range $[t_{v_s}, t_{w_s}]$. If the driver takes a detour to serve a rider as shown in Fig. 3, feasibility of the joint trip includes two conditions on their separate trip plans. The first condition is that the joint trip should have a positive TTS. Since a user can benefit from the ride-share only when the travel time of the joint trip $t_{v_i,v_s} + t_{v_s,w_s} + t_{w_s,w_i}$ is shorter than the sum of separate travel time values $t_{v_i,w_i} + t_{v_s,w_s}$, the positive TTS $\eta_{i,s}$ is necessary for users. It can be expressed as follows:

$$\begin{aligned} &t_{v_i,v_s} + t_{v_s,w_s} + t_{w_s,w_i} \leqslant t_{v_i,w_i} + t_{v_s,w_s}, \\ \Longleftrightarrow &\eta_{i,s} = t_{v_i,w_i} - (t_{v_i,v_s} + t_{w_s,w_i}) \geqslant 0. \end{aligned} \tag{4}$$

The second condition is that the joint trip should be time feasible. Only when both users' travel time schedules are respected, it is possible for the driver and the rider to share a trip. The confliction of users' time ranges can be expressed by the confliction of the joint trip's earliest departure time and the latest departure time. If the former is earlier than the latter, the joint trip is time feasible, and vice versa, i.e.,

$$\begin{aligned} &max(t_{v_i}, t_{v_s} - t_{v_i,v_s}) \leqslant min(t_{w_s} - t_{v_i,v_s} - \\ &t_{v_s,w_s}, t_{w_i} - t_{v_s,w_s} - t_{v_i,v_s} - t_{w_s,w_i}). \end{aligned} \tag{5}$$

The Equ. 5 is equal to the following inequations:

$$t_{v_i} \leqslant t_{w_s} - t_{v_i,v_s} - t_{v_s,w_s}, \tag{6}$$

$$t_{v_s} - t_{v_i,v_s} \leqslant t_{w_i} - t_{v_s,w_s} - t_{v_i,v_s} - t_{w_s,w_i}, \tag{7}$$

$$t_{v_i} \leqslant t_{w_i} - t_{v_s,w_s} - t_{v_i,v_s} - t_{w_s,w_i}, \tag{8}$$

$$t_{v_s} - t_{v_i,v_s} \leqslant t_{w_s} - t_{v_i,v_s} - t_{v_s,w_s}. \tag{9}$$

Note that the last inequation $t_{v_s} - t_{v_i,v_s} \leqslant t_{w_s} - t_{v_i,v_s} - t_{v_s,w_s} \Longleftrightarrow t_{v_s,w_s} \leqslant t_{w_s} - t_{v_s}$ holds due to Equ. (3). Hence, the RS-server only needs to test whether Equ. (6), (7) and (8) hold for the second condition of the feasible ride-share match.

For a feasible ride-share match, if TTS of the corresponding joint trip is larger, participants would save more travel cost and have a stronger incentive to accept the ride-share match [1].

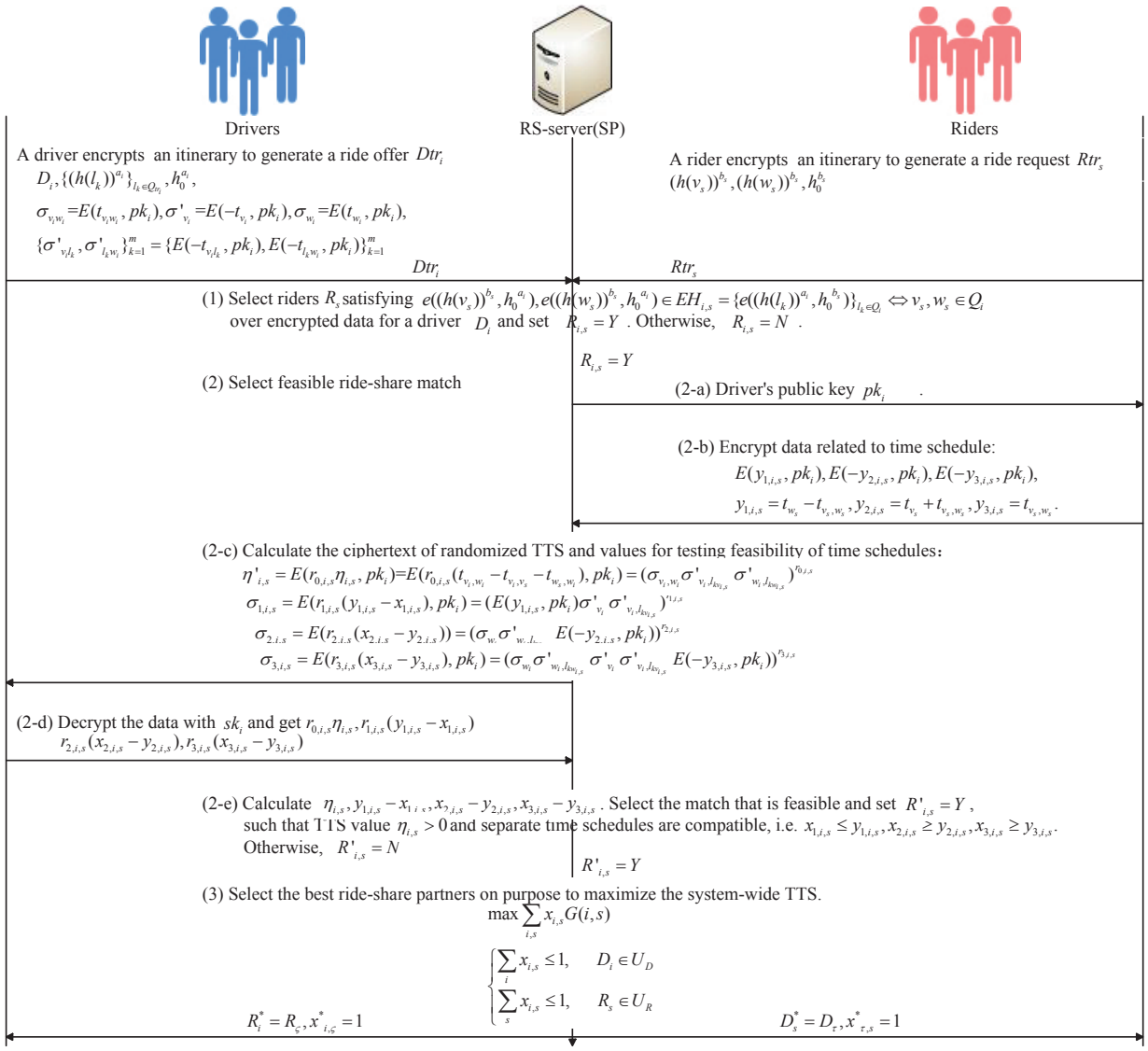


Fig. 4. PRIS

Thus, the feasible TTS based on map distance is able to act as the proximity between a driver and a rider in PRIS.

② Ride Offer/Request Generation

Both ride offer and ride request can be generated locally and offline.

Rider R_s encrypts public parameter h_0, her/his source v_s and destination w_s with a secret random integer $b_s \in Z_q^*$ by calculating $\{h_0^{b_s}, (h(v_s))^{b_s}, (h(w_s))^{b_s}\}$. Then, rider R_s forms ride request $Rtr_s = \{rider\, R_s, h_0^{b_s}, (h(v_s))^{b_s}, (h(w_s))^{b_s}\}$ and sends Rtr_s to the RS-server.

Driver D_i hides her/his acceptable spatial region Q_i by calculating $\{(h(l_k))^{a_i}\}_{l_k \in Q_i}$ and calculates $h_0^{a_i}$ with a secret random integer $a_i \in Z_q^*$. For location pairs (v_i, l_k) and (l_k, w_i), $l_k \in Q_i$, driver D_i encrypts the opposite of the expected vehicle time $-t_{v_i,l_k} \in Z_N$ and $-t_{l_k,w_i} \in Z_N$ with D_i's public key of Paillier cryptosystem pk_i to get $\sigma'_{v_i,l_k} = E(-t_{v_i,l_k}, pk_i)$ and $\sigma'_{w_i,l_k} = E(-t_{l_k,w_i}, pk_i)$. Other trip information including t_{v_i,w_i}, t_{v_i} and t_{w_i} is also encrypted to get $\sigma_{v_i,w_i} = E(t_{v_i,w_i}, pk_i)$, $\sigma'_{v_i} = E(-t_{v_i}, pk_i)$ and $\sigma_{w_i} = E(t_{w_i}, pk_i)$. Then, driver D_i sends ride offer $Dtr_i = \{driver\, D_i, \{(h(l_k))^{a_i}, \sigma'_{v_i,l_k}, \sigma'_{w_i,l_k}\}_{l_k \in Q_i}, h_0^{a_i}, \sigma_{v_i,w_i}, \sigma'_{v_i}, \sigma_{w_i}\}$ to the RS-server.

③ Three-Step Ride Selection.

In this subsection, we present how the RS-server communicates with drivers and riders to select potential feasible ride-share partners from all known users. For any pair of driver D_i and rider R_s, PRIS includes the following three steps as shown in Fig. 4.

(1) Spatial Region-based Pre-selection

To reduce the time cost of comparison calculation in the next step, the RS-server preselects riders whose pick-up and drop-off points are close to the driver's route, i.e. $v_s, w_s \in Q_i$, for driver D_i. The RS-server calculates $EH_{i,s} = \{e((h(l_k))^{a_i}, h_0^{b_s})\}_{l_k \in Q_i} = \{e(h(l_k), h_0)^{a_i b_s}\}_{l_k \in Q_i}$ for a pair of driver D_i and rider R_s. Considering

$$\begin{aligned} & EH_{i,s} = \{e(h(l_k), h_0)^{a_i b_s}\}_{l_k \in Q_i}, \\ & e((h(v_s))^{b_s}, h_0^{a_i}) = e(h(v_s), h_0)^{a_i b_s} \in EH_{i,s}, \\ & e((h(w_s))^{b_s}, h_0^{a_i}) = e(h(w_s), h_0)^{a_i b_s} \in EH_{i,s} \\ \Longleftrightarrow & h(v_s), h(w_s) \in \{h(l_k)\}_{l_k \in Q_i} \\ \Longleftrightarrow & v_s, w_s \in Q_i, \end{aligned}$$

the RS-server preselects rider R_s for driver D_i and sets

$R_{i,s}$ = "Y" if and only if $e(h(v_s),h_0)^{a_i b_s} \in EH_{i,s}$ and $e(h(w_s),h_0)^{a_i b_s} \in EH_{i,s}$. Otherwise, $R_{i,s}$ = "N". If $R_{i,s}$ = "Y", there certainly exists the numbers $kv_{i,s}$ and $kw_{i,s}$, such that $l_{kv_{i,s}} = v_s$ and $l_{kw_{i,s}} = w_s$.

(2) Feasible TTS-based Ride Matching

The RS-server further selects feasible pairs of drivers and riders over received encrypted trip data. The feasible ride-share match has a positive TTS as shown in Equ. (4), and is time feasible as expressed in (6), (7) and (8). The feasible TTS-based ride matching is performed as follows.

(2-a) For $R_{i,s}$ = "Y", the RS-server sends D_i's name to rider R_s.

(2-b) Rider R_s uses her/his own trip plan tr_s to calculate $y_{1,i,s} = t_{w_s} - t_{v_s,w_s}$, $y_{2,i,s} = t_{v_s} + t_{v_s,w_s}$ and $y_{3,i,s} = t_{v_s,w_s}$ for driver D_i. Then, the rider encrypts $\{y_{1,i,s}, -y_{2,i,s}, -y_{3,i,s}\}$ with the driver's public key pk_i to get $\{E(y_{1,i,s},pk_i), E(-y_{2,i,s},pk_i), E(-y_{3,i,s},pk_i)\}$ and sends the ciphertext to the RS-server.

(2-c) To prepare for the positive TTS selection expressed in the Equ. (4), the RS-server chooses a random number $r_{0,i,s} \in Z_N$ to calculate $\eta'_{i,s}$ based on the homomorphism of Paillier cryptosystem,

$$\begin{aligned}\eta'_{i,s} =& E(r_{0,i,s}\eta_{i,s}, pk_i),\\ =& (E(t_{v_i,w_i} - t_{v_i,v_s} - t_{w_s,w_i}, pk_i))^{r_{0,i,s}}\\ =& (E(t_{v_i,w_i}, pk_i)E(-t_{v_i,l_{kv_{i,s}}}, pk_i)E(-t_{l_{kw_{i,s}},w_i}, pk_i))^{r_{0,i,s}}\\ =& (\sigma_{v_i,w_i}\sigma'_{v_i,l_{kv_{i,s}}}\sigma'_{w_i,l_{kw_{i,s}}})^{r_{0,i,s}}.\end{aligned}$$

The RS-server sets $x_{1,i,s} = t_{v_i} + t_{v_i,v_s}$, $x_{2,i,s} = t_{w_i} - t_{w_s,w_i}$ and $x_{3,i,s} = t_{w_i} - t_{v_i} - t_{v_i,v_s} - t_{w_s,w_i}$. Then, the problem of testing time feasibility shown in Equ. (6), (7), and (8) can be transformed into the problem of comparing integers $x_{j,i,s}$ and $y_{j,i,s}$, $j = 1,2,3$. To solve this integers comparison problem without learning $\{x_{j,i,s}, y_{j,i,s}\}_{j=1}^3$, the RS-server calculates the encrypted $r_{1,i,s}(y_{1,i,s} - x_{1,i,s})$, $r_{2,i,s}(x_{2,i,s} - y_{2,i,s})$ and $r_{3,i,s}(x_{3,i,s} - y_{3,i,s})$ with driver D_i's public key pk_i based on the homomorphism and self-blinding property of Paillier cryptosystem, denoted by $\{\sigma_{1,i,s}, \sigma_{2,i,s}, \sigma_{3,i,s}\}$ as follows:

$$\begin{aligned}\sigma_{1,i,s} &= E(r_{1,i,s}(y_{1,i,s} - x_{1,i,s}), pk_i),\\ &= (E(y_{1,i,s}, pk_i)E(-t_{v_i} - t_{v_i,v_s}, pk_i))^{r_{1,i,s}},\\ &= (E(y_{1,i,s}, pk_i)\sigma'_{v_i}\sigma'_{v_i,l_{kv_{i,s}}})^{r_{1,i,s}},\end{aligned}$$

$$\begin{aligned}\sigma_{2,i,s} &= E(r_{2,i,s}(x_{2,i,s} - y_{2,i,s}), pk_i)\\ &= (E(t_{w_i} - t_{w_s,w_i}, pk_i)E(-y_{2,i,s}, pk_i))^{r_{2,i,s}}\\ &= (\sigma_{w_i}\sigma'_{w_i,l_{kw_{i,s}}}E(-y_{2,i,s}, pk_i))^{r_{2,i,s}},\end{aligned}$$

$$\begin{aligned}\sigma_{3,i,s} &= E(r_{3,i,s}(x_{3,i,s} - y_{3,i,s}), pk_i)\\ &= (E(t_{w_i}, pk_i)E(-t_{w_s,w_i}, pk_i)E(-t_{v_i}, pk_i)\\ &\quad E(-t_{v_i,v_s}, pk_i)E(-y_{3,i,s}, pk_i))^{r_{3,i,s}}\\ &= (\sigma_{w_i}\sigma'_{w_i,l_{kw_{i,s}}}\sigma'_{v_i}\sigma'_{v_i,l_{kv_{i,s}}}E(-y_{3,i,s}, pk_i))^{r_{3,i,s}},\end{aligned}$$

where random integers $r_{1,i,s}, r_{2,i,s}, r_{3,i,s} \in Z_N$. The RS-server sends $\eta'_{i,s}, \{\sigma_{j,i,s}\}_{j=1}^3$ to driver D_i.

(2-d) Driver D_i uses her/his secret key sk_i to decrypt the received ciphertexts and get the randomized value: $C = \{r_{0,i,s}\eta_{i,s}, r_{1,i,s}(y_{1,i,s} - x_{1,i,s}), r_{2,i,s}(x_{2,i,s} - y_{2,i,s}), r_{3,i,s}(x_{3,i,s} - y_{3,i,s})\}$. The driver sends C to the RS-server.

(2-e) The RS-server removes random integers $\{r_{0,i,s}, r_{1,i,s}, r_{2,i,s}, r_{3,i,s}\}$ from C and obtains $\eta_{i,s}$, $y_{1,i,s} - x_{1,i,s}$, $x_{2,i,s} - y_{2,i,s}$ and $x_{3,i,s} - y_{3,i,s}$. Since each value in $\{\eta_{i,s}, y_{1,i,s} - x_{1,i,s}, x_{2,i,s} - y_{2,i,s}, x_{3,i,s} - y_{3,i,s}\}$ represents a time range in a day, its value is within $[-L, L]$, which is transformed into $[0, L] \cup [L, N-L] \in Z_N$, where $|N| = 1024$ bits. If $\eta_{i,s}, y_{1,i,s} - x_{1,i,s}, x_{2,i,s} - y_{2,i,s}, x_{3,i,s} - y_{3,i,s} \in [0, L]$ holds in Z_N, all of inequalities Equ. (4), (6), (7) and (8) hold, i.e. $\eta_{i,s} \geq 0$, $x_{1,i,s} \leq y_{1,i,s}$, $x_{2,i,s} \geq y_{2,i,s}$ and $x_{3,i,s} \geq y_{3,i,s}$, and vice versa. Thus, if $\eta_{i,s}, y_{1,i,s} - x_{1,i,s}, x_{2,i,s} - y_{2,i,s}, x_{3,i,s} - y_{3,i,s} \in [0, L]$, the ride-share match between driver D_i and rider R_s is feasible, and the RS-server sets $R'_{i,s}$ = "Y". Otherwise, $R'_{i,s}$ = "N".

(3) Maximal System-wide TTS Optimization

The RS-server finally selects the best ride-share partners on the purpose to maximize the system-wide TTS.

The RS-server creates a weight bipartite G based on selected feasible ride-share candidates. If $R'_{i,s}$ = "N", it is impossible for driver D_i to share a trip with rider R_s, i.e. $G(i,s) = 0$. If $R'_{i,s}$ = "Y", an edge connecting driver node D_i and rider node R_s with the weight $G(i,s) = \eta_{i,s}$, indicating that driver D_i and rider R_s can share a ride with potential TTS value $\eta_{i,s}$. Therefore, the RS-server formularizes the maximal system-wide TTS optimization problem to select a ride-share partner for every user as shown in Equ. (10).

$$\begin{aligned}&max \sum_{i,s} x_{i,s}G(i,s)\\ &\begin{cases}\sum_i x_{i,s} \leq 1 & \text{all riders } R_s \in U_R,\\ \sum_s x_{i,s} \leq 1 & \text{all drivers } D_i \in U_D,\end{cases}\end{aligned} \tag{10}$$

where a decision variable $x_{i,s} = 0$ or 1 corresponds to a ride-share of driver D_i and rider R_s. If $x_{i,s} = 1$, the RS-server recommends driver D_i and rider R_s to share a ride. The constraints force each rider to be a passenger of at most one driver and vice versa.

The maximum weighted bipartite optimal matching problem can be efficiently solved in polynomial time $O(n_0 {n_1}^2)$ by a modified version of the KM algorithm [39], where $n_0 = min(|U_D|, |U_R|)$, $n_1 = max(|U_D|, |U_R|)$. Assuming the result of optimization problem in Equ. (10) is denoted by $x^*_{i,s}$. Driver D_i's and rider R_s's best matches are

$$R^*_i = \{R_\varsigma | x^*_{i,\varsigma} = 1\}, D^*_s = \{D_\tau | x^*_{\tau,s} = 1\},$$

respectively. The RS-server sends the best ride-share partner's name R^*_i and D^*_s to driver D_i and rider R_s, respectively. It enables users to communicate with each other and enjoy a ride-sharing trip with a low total time cost.

V. Security Analysis

In this section, we analyze the security of this proposed scheme to show the achievement of its security goals.

For a given pair of driver D_i and rider R_s, RS-server obtains ride request $Rtr_s = \{R_s, h_0^{b_s}, (h(v_s))^{b_s}, (h(w_s))^{b_s}\}$, ride offer $Dtr_i = \{D_i, \{(h(l_k))^{a_i}, \sigma'_{v_i,l_k}, \sigma'_{w_i,l_k}\}_{l_k \in Q_i}, h_0^{a_i}, \sigma_{v_i,w_i}, \sigma'_{v_i}, \sigma_{w_i}\}$, ciphertexts for TTS calculation $\{E(y_{1,i,s}, pk_i), E(-y_{2,i,s}, pk_i), E(-y_{3,i,s}, pk_i)\}$, TTS of the shared trip $\eta_{i,s} = t_{v_i,w_i} - t_{v_i,v_s} - t_{w_s,w_i}$ and some integers used in the second selection step $y_{1,i,s} - x_{1,i,s}$, $x_{2,i,s} - y_{2,i,s}$ and $x_{3,i,s} - y_{3,i,s}$.

The RS-server tests whether rider R_s's source and destination are within the driver's spatial region Q_i or not by testing

TABLE III
COMPLEXITY ANALYSIS

Our scheme		Our scheme			ORide [21]			EP [33]	
Entity		Driver	Rider	RS-server	Driver	Rider	RS-server	Driver	Rider
Offline Comp.		$m\,h$, $(2m+3)\,mul_2$ $(7m+7)\,exp_1$	$3\,h$ $3\,exp_1$	- -	SHE.gen -	- -	- -	$3mul_2$ $6exp_1$	- -
Online Comp.		$4\,exp_2$	$3\,mul_2$ $6\,exp_1$	$4\,exp_2$, $8\,mul_2$, $4\,div$, $(m+2)\,bli$	SHE.enc	SHE.enc SHE.dec	SHE.$dist$ -	$r^2\,exp_2$ -	$(r^2+4)\,mul_2$, $2\,exp_1$ $(r^2+2)\,exp_2$
Comm. trans. (bits)	server	$(5m+11)\times 1024$	6×1024	-	248×10^3	372×10^3		-	-
	driver	-	-	8×1024	-	-	124×10^3	6×1024	-
	rider	-	-	1024	-	-	186×10^3	-	$2r^2\times 1024$

whether both $e((h(v_s))^{b_s}, h_0^{a_i}) \in EH_{i,s}$ and $e((h(w_s))^{b_s}, h_0^{a_i}) \in EH_{i,s}$ are true or not, where $EH_{i,s} = \{e((h(l_k))^{a_i}, h_0^{b_s})\}_{l_k \in Q_i}$. In PRIS, the RS-server does not need other information in addition to ciphertexts $(h(v_s))^{b_s}, (h(w_s))^{b_s}, h_0^{b_s}$, $(h(l_k))^{a_i}$ and $h_0^{a_i}$. Given the decisional Diffie-Hellman assumption, $\langle h_0, h_0^{b_s}, h(v_s), (h(v_s))^{b_s}\rangle$ and $\langle h_0, h_0^{b_s}, h(v_s), M\rangle$ are indistinguishable [40] for $b_s \in Z_q^*$, $h_0, h(v_s), M \in \mathbb{G}$. Thus, it is difficult for the RS-server to distinguish a given rider's $h(v_s)$ from other hash values. Similarly, the RS-server hardly deduces the rider's $h(w_s)$ and the driver's $h(l_k)$.

In the process of feasible TTS-based ride matching, driver D_i infers neither TTS of the joint trip $\eta_{i,s} = t_{v_i,w_i} - t_{v_i,v_s} - t_{w_s,w_i}$ nor the intermediate integers used to test feasibility of time schedules $y_{1,i,s} - x_{1,i,s}$, $x_{2,i,s} - y_{2,i,s}$ and $x_{3,i,s} - y_{3,i,s}$, since the random numbers $r_{0,i,s}, r_{1,i,s}, r_{2,i,s}$ and $r_{3x,i,s}$ are added into the received ciphertexts by using the self-blinding property of Paillier cryptosystem. Although the RS-server knows $\eta_{i,s}, y_{1,i,s} - x_{1,i,s}, x_{2,i,s} - y_{2,i,s}$ and $x_{3,i,s} - y_{3,i,s}$, there are nearly L^3 solutions of the unknown vector $(t_{v_i}, t_{w_i}, t_{v_i,v_s}, t_{w_s,w_i}, t_{v_s,w_s}, t_{w_s}, t_{v_s})$ in theory of algebraic equations and the RS-server cannot distinguish the right one from them.

AS1. For large-scale inference attacks launched by a RS-server, the RS-server needs to learn their identities, locations, route and time associated with trips to profile individual users' activities. From the above analysis, a RS-server can collect TTS and variables for testing compatibility of time schedules in a long period, i.e. $\eta_{i,s}$ and $\{y_{1,i,s} - x_{1,i,s}, x_{2,i,s} - y_{2,i,s}, x_{3,i,s} - y_{3,i,s}\}$ corresponding to driver D_i and rider R_s. The RS-server is able to learn when the driver and the rider share a ride with how much TTS, but the RS-server cannot infer exact source, destination, time schedule and route impacted in time schedule due to commutative encryption function and Paillier cryptosystem.

AS2. Assuming a RS-server is a stronger adversary. The RS-server knows the precise pick-up location v_s and a specific rider's time schedule $[t_{v_s}, t_{w_s}]$ and wants to know the drop-off location and time of the joint ride, i.e. w_s and $max(t_{v_i}, t_{v_s} - t_{v_i,v_s})$. In this case, the precise v_s and $[t_{v_s}, t_{w_s}]$ are not enough to get exact $\{t_{v_i}, t_{v_i,v_s}\}$ from values $\{\eta_{i,s}, y_{1,i,s} - x_{1,i,s}, x_{2,i,s} - y_{2,i,s}, x_{3,i,s} - y_{3,i,s}\}$, and vice versa. Hence, the proposed selection scheme can prevent riders from attack AS2.

AU. The information driver D_i knows about a given rider R_s includes intermediate values $r_{0,i,s}\eta_{i,s}$, $r_{1,i,s}(y_{1,i,s} - x_{1,i,s})$, $r_{2,i,s}(x_{2,i,s} - y_{2,i,s})$ and $r_{3,i,s}(x_{3,i,s} - y_{3,i,s})$. Since $r_{0,i,s}$, $r_{1,i,s}$, $r_{2,i,s}$ and $r_{3,i,s}$ are secret random numbers chosen by the RS-server, driver D_i has no way to infer information $\eta_{i,s}$, t_{v_i,v_s} and t_{w_s,w_i} related to rider R_s. Even if driver D_i strategically and frequently inputs ride offers with different locations and times on the purpose to infer a targeted rider's sensitive location, driver D_i cannot succeed to learn precise $\eta_{i,s}$, t_{v_i,v_s} and t_{w_s,w_i}, since the RS-server chooses new secret random numbers $\{r_{0,i,s}, r_{1,i,s}, r_{2,i,s}, r_{3,i,s}\}$ every time it sends message to driver D_i.

The information rider R_s receives and gets about a given driver D_i during feasible matching is the driver's public key in Paillier cryptosystem, so the rider totally has no idea about driver's sensitive location data.

In summary, PRIS can achieve the privacy protection goals presented in Sec.III-C for feasible ride-share partner selection in RSSs.

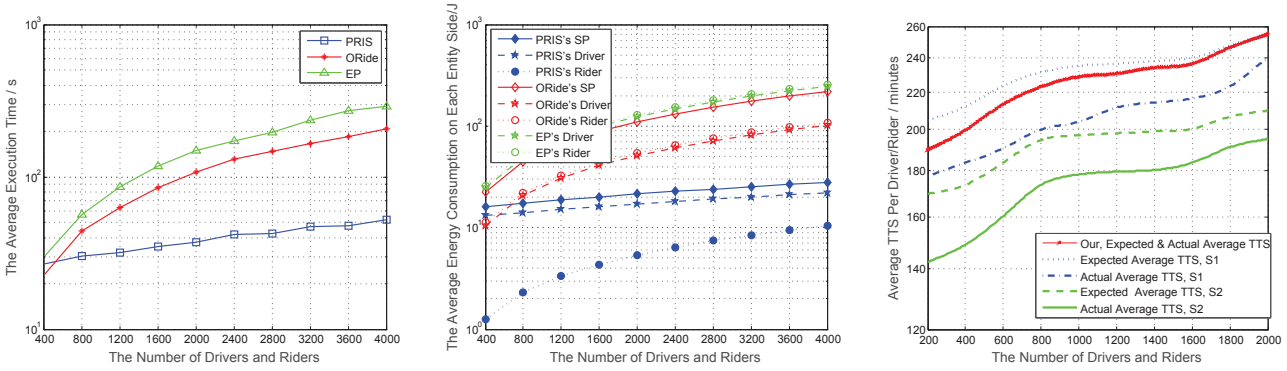
VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of PRIS through complexity analysis and simulation results, in terms of execution time, energy consumption and the selection rules. We use real ride-sharing data from the BlaBlaCar website for our simulations, consisting of 713,045 rides of 58,231 users. We assume that N and g are of 1024 and 160 bits, respectively, for the sufficient semantic security of Paillier Cryptosystem [38]. The length of each element of $\mathbb{G}$ is $|q| = 160$ bits, and the length of an element of $\mathbb{G}_T$ is 1024 bits for the bilinear map: $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

A. Complexity Analysis

Table III shows the comparison result of PRIS, Oride [21] and EP [33] in details, where m and L (minutes) in PRIS are the maximum size of driver's spatial region Q_i and the maximum TTS brought by a shared trip, respectively, and r (km) in EP [33] is the distance threshold that a driver allows deviating the planned route.

The computational cost is measured by counting the different operations. h represents a keyed hash function of SHA-1. exp_1, exp_2, mul_1, mul_2 and bli denote 1024-bit exponentiation, 2048-bit exponentiation, 1024-bit multiplication, 2048-bit multiplication and bilinear map in a cyclic multiplicative group (on an MNT curve with a base field size of 159 bits), respectively. SHE.gen, SHE.enc and SHE.dec represent key generation, encryption and decryption with coefficient of size 124 bits in SHE [21], NFLlib. The offline computational overhead refers to the computation cost of operations that



(a) (b) (c)

Fig. 5. Impact of the number of drivers and riders ($|U_D| + |U_R|$) on # (a): execution time; (b): energy consumption; (c): average TTS per driver/rider, on smartphones and server.

are executed offline. In the PRIS, offline operations are the operations carried out locally to generate drivers' ride offers and riders' ride requests, while the generation of drivers' ride offers in EP [33] is offline operation. The online part of PRIS starts with sending ride offers/requests to the RS-server and initiating the three-step ride selection. In EP [33], the online operations start when the driver sends a ride offer to a rider. As shown in Table III, each entity's online computation cost is constant in PRIS, ORide [21] and EP [33]. The online computation cost of EP [33] is the highest of all due to r^2 times of 2048-bit exponentiation and multiplication operations. Compared to PRIS, ORide [21] has the relatively heavy online computation overhead due to the encryption/decryption operations of SHE.

The communication cost is evaluated by counting the transmitted and received bits. In the ORide and PRIS, a driver's/rider's communication cost is measured by the number of bits that she/he transmits to the RS-server, since they don't communicate with each other until a successful establishment of ride-share match. In EP [33], we evaluate a driver's/rider's communication cost by counting the number of bits transmitted to another entity. The server's communication cost is the amount of bits sent to drivers and riders. Table III illustrates that the communication cost of the PRIS and EP [33] is higher than that of ORide [21]. The ride offer includes a lot of ciphertexts in the PRIS and EP [33], leading to high communication cost, but the transmission rate of Bluetooth v4.0 can exceed 900Kb/s easily, so the communication cost impacts less on the execution time.

B. Simulation Result

We implement PRIS on many Nexus 5 smartphones with 2.3 GHz CPU, 2G RAM, 16G ROM, Android 6.0 system, Bluetooth v4.0, and a DELL R720 server with Xeon E5-2690, 2.9GHz, 32GB RAM, OS Ubuntu server 14.04.

We choose 10 random days and extract 10 subsets from the collected data per day to run PRIS. Since a user can use several pseudonyms in PRIS and one pseudonym usually corresponds to one ride offer/request in one day, we assume that the number of drivers and riders is equal to the number of ride offers and ride requests in a subset. In these extracted subsets, the number of drivers and riders varies from 400 to 4000. Assuming that the size of driver's spatial region is $m = 100$ and the maximum TTS is $L = 1440$ (minutes) for any pairs of driver and rider in PRIS, and the distance threshold of detour is $r = 10$ (km) in EP [33]. We conduct extensive simulation based on all data sets and analyze the simulation results of serving different numbers of users. Since all users can perform the scheme simultaneously in practical RSSs applications, we analyze the simulation results in the aspects of average execution time, average energy consumption and average TTS per driver/rider, instead of total execution time, energy consumption and TTS, respectively. The total execution time refers to how much time the server and all users cooperate to conduct the scheme and get a ride matching result, so the average execution time per user is equal to the total execution time over the number of drivers and users, as well as average energy consumption and average TTS per driver/rider. For briefness, we omit "average" and "per driver/rider" in the following passage.

Execution time. Fig. 5(a) illustrates that the execution time of ORide [21], EP [33] and PRIS with the increasing size of user set. The PRIS is much more efficient than EP [33] and ORide [21] if the number of drivers and riders is more than 500, since the PRIS has the lower online computation cost. For each driver/rider, ORide [21] and EP [33] needs 107.83s and 245.48s to find her/his best ride-share partner from 2000 users, respectively, while the PRIS requires 37.13s to serve the user.

Energy consumption. The energy consumption model [41] $E_{computing} = P_{comp}T_{comp} + 0.3167T_{run}$ is used to estimate local computational cost. The energy consumption model of network transmission cost is based on [42]: $E_{network} = n_t E_t + n_r E_r$. Thus, we evaluate the energy consumption according to: $E = E_{computing} + E_{network}$. Fig. 5(b) shows each participant's energy consumption for finding a user's ride-share partner in ORide [21], PRIS and EP [33]. When 2000 users send ride offers/requests simultaneously, PRIS needs to consume 17.06J on the driver side, 5.37J on the rider side and 21.26J on the server side to finish ride-share partner selection for any user, while these values in ORide [21] are 51.20J, 54.45J and 109.48J, and EP [33] consumes 123.56J on the driver and 128.96J on the rider side, respectively. The RS-server

consumes less energy in PRIS than it does in ORide [21], and either a driver or a rider consumes less energy in PRIS than she/he does in ORide [21] and EP [33] as well.

Effect of selection rules on TTS. To show the advantage of our three-step partner selection, we compare the effects of three different selection rules on the average TTS per driver/rider, including rules considering (**Our**) map distance-based TTS, (**S1**) Euclidean distance-based TTS and (**S2**) Euclidean distance-based greedy ride-share matching. The average TTS per driver/rider is defined as $a_{TTS} = \frac{\sum TTS}{|U_D|+|U_R|}$, where $\sum TTS$ and $|U_D|+|U_R|$ denote the system-wide TTS and the number of drivers and riders, respectively. Fig. 5(c) shows that the more users participate in ride matching, the larger average TTS per driver/rider can obtain, whatever selection rules are followed.

If a scheme evaluates a_{TTS} based on Euclidean distance rather than map distance, there would exist a gap between the expected a_{TTS} and the actual value. Hence, the expected a_{TTS} is bigger than the actual a_{TTS} in both **S1** and **S2** as shown in Fig. 5(c). In the PRIS, the actual a_{TTS} is equal to the expected value due to map distance. Besides, **S2** offers the lowest average TTS, since the greedy rule, i.e. selecting the nearest rider/driver for a driver/rider, cannot provide the globally optimal TTS. Finally, **S1** has the highest expected a_{TTS} per driver/rider in Fig. 5(c). The reason is that Euclidean distance-based driving time from a location to another is usually shorter than map distance-based value. This fact enhances the possibility of selecting matched pairs of users under their time constraints, and thereby leads to the highest expected a_{TTS}. Unfortunately, the expected a_{TTS} does not comport with the actual value, since the imprecise distance measurement leads to some conflicts between selected users' time ranges.

VII. Conclusions

In this paper, we have proposed PRIS, a privacy-preserving partner selection scheme for RSSs, incorporating the positive TTS and the feasibility of separate time schedules without exposing the concrete itineraries of both riders and drivers. Specifically, PRIS enables the RS-server to select the proper rider matching based on the spatial regions, positive TTS and system-wide TTS, ensuring that the pairs of drivers and riders for ride-sharing are optimal on physical regions, time schedules and travel time saving. The trip data is protected during partner selection to prevent the curious RS-server and other users from invading the privacy of both riders and drivers. In the future work, we will design a privacy-preserving partner selection scheme considering the trust levels of drivers and riders in RSSs.

Acknowledgment

We would like to thank the anonymous reviewers for their helpful comments. This work is partially supported by the grants from the National Key R&D Program of China (2017YFB0802203) and the National Natural Science Foundation of China (61672515, 61502489).

References

[1] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295–303, 2012.

[2] Y. Hou, W. Zhong, L. Su, K. Hulme, A. W. Sadek, and C. Qiao, "Taset: Improving the efficiency of electric taxis with transfer-allowed rideshare," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9518–9528, 2016.

[3] F. Qu, Z. Wu, F. Y. Wang, and W. Cho, "A security and privacy review of vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2985–2996, 2015.

[4] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *Proc. of IEEE INFOCOM*, 2012, pp. 729–737.

[5] Futrism, "Uber releases a staggering 2 billion trips-worth of traffic data," https://futurism.com/uber-releases-a-staggering-2-billion-trips-worth-of-traffic-data/.

[6] N. Perlroth, "All 3 billion yahoo accounts were affected by 2013 attack," https://www.nytimes.com/2017/10/03/technology/yahoo-hack-3-billion-users.html.

[7] Wikipedia, "icloud leaks of celebrity photos," https://en.wikipedia.org/wiki/ICloud_leaks_of_celebrity_photos.

[8] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for internet of things applications: challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, pp. 1–28, 2017.

[9] D. Coldewey, "Uber fined in settlement with NY over 'god view' tracking," http://www.nbcnews.com/tech/tech-news/uber-fined-settlement-ny-over-god-view-tracking-n491706.

[10] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in vanets," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 86–96, 2012.

[11] J. Ni, K. Zhang, X. Lin, H. Yang, and X. S. Shen, "Ama: Anonymous mutual authentication with traceability in carpooling systems," in *Proc. of IEEE ICC*, 2016, pp. 1–6.

[12] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–12, 2017.

[13] Y. Sun, R. Lu, X. Lin, X. Shen, and J. Su, "An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 7, pp. 3589–3603, 2015.

[14] P. Goel, L. Kulik, and K. Ramamohanarao, "Privacy-aware dynamic ride sharing," *ACM Transactions on Spatial Algorithms and Systems*, vol. 2, no. 1, pp. 4:1–4:41, 2016.

[15] M. A. Ferrag, L. Maglaras, and A. Ahmim, "Privacy-preserving schemes for ad hoc social networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, pp. 1–33, 2016.

[16] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proc. of IEEE INFOCOM*, 2014, pp. 754–762.

[17] A. Pham, I. Dacosta, B. Jacotguillarmod, K. Huguenin, T. Hajar, F. TramÍlr, V. Gligor, and J. P. Hubaux, "Privateride: A privacy-enhanced ride-hailing service," in *Proc. of PETS*, 2017, pp. 38–56.

[18] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics & Security*, vol. 11, no. 12, pp. 2706–2716, 2017.

[19] C. Dong, L. Chen, and Z. Wen, "When private set intersection meets big data: an efficient and scalable protocol," in *Proc. of CCS*, 2013, pp. 789–800.

[20] X. Lin, R. Lu, X. Liang, and X. Shen, "Stap: A social-tier-assisted packet forwarding protocol for achieving receiver-location privacy preservation in vanets," in *Proc. of IEEE INFOCOM*, 2011, pp. 2147–2155.

[21] A. Pham, I. Dacosta, G. Endignoux, J. R. T. Pastoriza, K. Huguenin, and J.-P. Hubaux, "ORide: A privacy-preserving yet accountable ride-hailing service," in *Proc. of USENIX*, 2017, pp. 1235–1252.

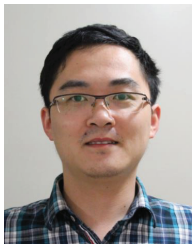
[22] J. Friginal, S. Gambs, J. Guiochet, and M. O. Killijian, "Towards privacy-driven design of a dynamic carpooling system," *Pervasive & Mobile Computing*, vol. 14, pp. 71–82, 2014.

[23] P. Goel, L. Kulik, and K. Ramamohanarao, "Optimal pick up pointselection for effective ride sharing," *IEEE Transactions on Big Data*, vol. 3, no. 99, pp. 154–168, 2017.

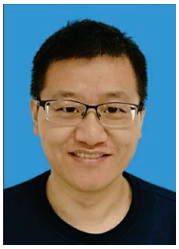
[24] B. Gedik and L. Liu, "Mobieyes: A distributed location monitoring service using moving location queries," *IEEE Transactions on Mobile Computing*, vol. 5, no. 10, pp. 1384–1402, 2006.

[25] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.
[26] R. Chen, B. C. M. Fung, N. Mohammed, B. C. Desai, and K. Wang, "Privacy-preserving trajectory data publishing by local suppression," *Information Sciences*, vol. 231, no. 1, pp. 83–97, 2013.
[27] M. P. Kwan, I. Casas, and B. C. Schmitz, "Protection of geoprivacy and accuracy of spatial information: How effective are geographical masks?" *Cartographica the International Journal for Geographic Information & Geovisualization*, vol. 39, no. 2, pp. 15–28, 2004.
[28] Y. Xi, L. Schwiebert, and W. Shi, "Privacy preserving shortest path routing with an application to navigation," *Pervasive & Mobile Computing*, vol. 13, no. 4, pp. 142–149, 2014.
[29] D. J. Wu, J. Zimmerman, J. Planul, and J. C. Mitchell, "Privacy-preserving shortest path computation," in *Proc. of NDSS*, 2016, pp. 1–15.
[30] I. Bilogrevic, M. Jadliwala, V. Joneja, and K. Kalkan, "Privacy-preserving optimal meeting location determination on mobile devices," *IEEE Transactions on Information Forensics & Security*, vol. 9, no. 7, pp. 1141–1156, 2014.
[31] M. Stiglic, N. Agatz, M. Savelsbergh, and M. Gradisar, "The benefits of meeting points in ride-sharing systems," *Transportation Research Part B Methodological*, vol. 82, pp. 36–53, 2015.
[32] F. Li, H. Li, Y. Jia, N. Yu, and J. Weng, "Privacy computing: concept, connotation and its research trend," *Journal on Communications*, vol. 37, no. 4, pp. 78:1–11, 2016.
[33] P. Hallgren, C. Orlandi, and S. Andrei, "Privatepool: privacy-preserving ridesharing," in *Proc. of IEEE CSF*, 2017, pp. 276–291.
[34] C. Dai, X. Yuan, and C. Wang, "Privacy-preserving ridesharing recommendation in geosocial networks," in *Proc. of CSoNet*, 2016, pp. 193–205.
[35] Y. He, K. Zhang, H. Wang, F. Li, B. Niu, and H. Li, "Impact factor-based group recommendation scheme with privacy preservation in msns," in *Proc. of ICC*, 2017, pp. 1938–1883.
[36] D. Perry, "Sex and uber's 'rides of glory': The company tracks your one-night stands – and much more," http://www.nbcnews.com/tech/tech-news/uber-fined-settlement-ny-over-god-view-tracking-n491706.
[37] D. Sánchez, S. Martínez, and J. Domingo-Ferrer, "Co-utile p2p ridesharing via decentralization and reputation management," *Transportation Research Part C*, vol. 73, pp. 147–166, 2016.
[38] R. Zhang, J. Zhang, Y. Zhang, J. Sun, and G. Yan, "Privacy-preserving profile matching for proximity-based mobile social networking," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 656–668, 2013.
[39] A. McGregor, "Graph stream algorithms: A survey," in *Proc. of SIGMOD*, 2014, pp. 9–20.
[40] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases," in *Proc. of SIGMOD*, 2003, pp. 86–97.
[41] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. of USENIX*, 2010, pp. 1–14.
[42] M. Naor, "Cryptography and mechanism design," in *Proc. of TRAK*, 2001, pp. 163–167.

Yuanyuan He (S'15) received the B.E. degree and the M.S. degree in Computational Mathematics from the Chongqing University, Chongqing, China, in 2006 and 2009, respectively. She is currently working toward the Ph.D. degree in State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences. She was a visiting scholar in University of Waterloo, Waterloo, ON, Canada. Her current research interests include wireless network security, privacy computing, fog computing and Internet of Things. (E-mail: heyuanyuan@iie.ac.cn)



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.



Xinyu Wang received his B.S. and M.S. degrees in Computer Science and Technoloy from Xidian University in 2011 and 2014, respectively. Currently, he is working toward the Ph.D. degree in State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences. His current research interests include wireless network security, privacy computing. (E-mail: wangxinyu@iie.ac.cn)

Ben Niu (corresponding author) received his B.S. degree in Information Security, M.S. and Ph.D. degrees in Cryptography from Xidian University in 2006, 2010 and 2014 respectively. Currently, he is working as research assistant in State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences. He was a visiting scholar in Pennsylvania State University from 2011 to 2013. His current research interests include wireless network security, privacy computing. (E-mail: niuben@iie.ac.cn)

Fenghua Li received his B.S. degree in Computer Software, M.S. and Ph.D. degrees in Computer Systems Architecture from Xidian University in 1987, 1990 and 2009 respectively. Currently, he is working as professor and doctoral supervisor in State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences. He is also a doctoral supervisor in Xidian University. His current research interests include network security, system security, privacy computing and trusted computing. (E-mail: lfh@iie.ac.cn)

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received Ph.D. degrees (1990) from Rutgers University, New Jersey (USA). Dr. Shen is a University Professor, Department of Electrical and Computer Engineering, University ofWaterloo, 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. He also serves as the Editor-in-Chief for IEEE Internet of Things Journal, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications. 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.