Location Privacy in Usage-Based Automotive Insurance: Attacks and Countermeasures

Location Privacy in Usage-based Automotive Insurance: Attacks and Countermeasures

Lu Zhou†, Suguo Du†, Haojin Zhu†, *Senior Member, IEEE,* Cailian Chen†, *Member, IEEE,* Kaoru Ota‡, *Member, IEEE,* and Mianxiong Dong ‡, *Member, IEEE*
† Shanghai Jiao Tong University, China ‡ Muroran Institute of Technology, Japan

***Abstract*—Usage-based insurance (UBI) is regarded as a promising way to provide accurate automotive insurance rates by analyzing the driving behaviors (e.g., speed, mileage, and harsh braking/accelerating) of drivers. The best practice that has been adopted by many insurance programs to protect users' location privacy is the use of driving speed rather than GPS data. However, in this study, we challenge this approach by presenting a novel speed-based location trajectory inference framework. The basic strategy of the proposed inference framework is motivated by the following observations. In practice, many environmental factors such as real-time traffic and traffic regulations, can influence the driving speed. These factors provide side-channel information about the driving route, which can be exploited to infer the vehicle's trace. We implement our discovered attack on a public dataset in New Jersey. The experimental results show that the attacker has a nearly 60% probability of obtaining the real route if he chooses the top 10 candidate routes. To thwart the proposed attack, we design a privacy preserving scoring and data audition Framework that enhances drivers' control on location privacy without affecting the utility of UBI. Our defense framework can also detect users' dishonest behavior (e.g. modification of speed data) via a probabilistic audition scheme. Extensive experimental results validate the effectiveness of the defense framework.**

***Index Terms*—Connected Vehicles, Location Privacy, Hidden Markov Model, Secure Aggregation Protocol, Inspection Game**

I. Introduction

THE current pricing policy of automotive insurance companies around the world is based on traditional factors, such as age, location of residence, history of accidents and traffic violations. This means that all customers pay similar prices for similar factors, despite potentially large variations in their driving habits. The emerging telematics-based usage-based insurance (or pay-how-you-drive programs) is dramatically reshaping the landscape of the global auto insurance market. Examples of such programs in North America and Europe include Progressive's Snapshot [2], AllState's Drivewise [3], State Farm's In-Drive [4], and Travelers' Intellidrive [5].

Usage-based insurance (UBI) relies on the collection of each driver's data using various technologies (OBD-II, Smartphone, or Hybrid OBD-Smartphone) to calculate the risk score during a monitoring period, which can reflect the probability of getting involved in an accident. UBI provides a promising way to differentiate safe drivers from risky ones, which forms the basis for risk categorization and, thus, for subsequent discounts or surcharges on premiums depending on driving behavior. The number of UBI market subscribers is expected to reach approximately 100 million by 2020, and UBI is projected to be used by approximately 50% of the world's vehicles by 2030 [6].

Although UBI is regarded as a promising approach for offering more accurate insurance services by profiling driving habits, the data that are collected via this method can compromise users' privacy, especially users' location privacy. Many insurance programs, which are advertised as being privacy-preserving, record the speed rather than directly using GPS-based tracking. Previous works [7] [8] challenge this best practice by proposing a tracking algorithm that is based on the collected speed data. Unfortunately, the proposed algorithms suffer from limited tracking performance. What is more important, how to design a privacy preserving UBI, which allows the insurance company to provide fine-grained insurance plans for drivers based on their driving habits without compromising their location privacy, still remains a great challenge and has received less attention so far.

To overcome the above research challenges, we investigate the location privacy problem in UBI from the following aspects: In our previous work [1], we proposed and designed a novel speed-based trajectory inference algorithm that can accurately track drivers based only on driving speed. The proposed algorithm is motivated by the following observations: Due to the development of various location-based services (LBSs), it is easy to automatically retrieve the road speed limit and real-time traffic information from publicly available interfaces that are provided by mainstream navigation systems such as Google and Baidu Map. These speed limits and real-time traffic information provide us with important information regarding the actual driving speed of the target vehicle on a specific road, which can be exploited by an attacker to infer the target vehicle's real trajectory based only on the speed data. We perform comprehensive experiments to evaluate the proposed attack, which shows significant improvement in term of the inference performance. We launch our attack with data from 120 trips and the experimental results show that the attacker has nearly 60% probability of obtaining the real route if he chooses the top 10 candidate routes of a trip.

Our previous work [1] only considered the question of how to infer drivers' trajectories, and how to design a privacy-preserving scoring framework without changing the current architecture has not been solved yet. Furthermore, from the

Part of this paper has been presented in the 37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017 (short paper track, six pages) [1].

Corresponding author: Haojin Zhu (email: zhu-hj@sjtu.edu.cn).

perspective of the insurer, it still remains a great challenge to verify the authenticity of the uploaded data. In this paper, to provide the privacy-preserving features and achieve secure UBI, we introduce a general scoring method (in section III-B) and propose a privacy-preserving scoring and data audition framework for UBI, which is denoted as Pri-UBI. The basic goal of Pri-UBI is to allow the driver to have enhanced control of his driving data while ensuring the scoring accuracy, and preventing him from forging data.

Pri-UBI is composed of two major modules: the private usage-based scoring (Pri-UBS) algorithm and the probabilistic usage data audition (Pro-UDA) protocol. Pri-UBS is designed to enable the insurer to calculate the risk of a driver based on his driving data without revealing the individual data, which is expected to significantly enhance the driver's control of his privacy while still allowing the insurer to rate the driver based on his driving habits. However, since these driving data are collected from the driver's smartphone, the driver has an incentive to modify them to obtain the benefits. Inspired by the Inspection game, we introduce a probabilistic usage data audition scheme that can probabilistically audit the driver's driving data at a reduced cost.

The main contributions of this paper are summarized as follows:

- We perform a comprehensive survey of the current UBI system and propose a general scoring model based on the existing industry practice and prior works.
- We propose a privacy preserving scoring and data audition framework for usage-based insurance, which can protect individual driving data while ensuring that insurance companies can rate the driver based on his driving habits. Our scheme can also prevent users from modifying data by probabilistic audition.
- We add detailed experiments to explore factors that potentially influence the performance of our trajectory inference system, which have not been discussed in the conference paper. Then, we design a comparative experiment to demonstrate the performance of our proposed inference system.
- We perform a detailed security analysis and experiment to demonstrate our proposed framework. The results demonstrate the proposed privacy-preserving framework is robust against malicious/selfish attackers.

The rest of this paper is organized as follows. Firstly, we introduce the related work in Section II. We then briefly introduce the current architecture and the scoring model of UBI in Section III. Section IV and Section V present the attack model and the novel trajectory inference attack system including problem formulation and the attack framework. Section VI gives the privacy-preserving scoring and data audition framework. Then we give the experimental evaluation in Section VII based on an inference system implementation on a public dataset in New jersey. Finally, we conclude the paper in Section VIII.

II. Related Work

In this section, we will introduce existing works on usage-based insurance from the perspective of the architecture and scoring models, security and privacy issues, and countermeasures.

A. The Architecture and scoring models of UBI

Nowadays, UBI has received the wide attentions from both of the academia and the industry. In [9], the authors proposed an overall smartphone measurement system model, spanning from the physical layer to the business model at the top layer. Some existing works [10], [11] proposed many scoring models for providing a more reasonable scoring for UBI. Among the different models, Pay-How-you-Drive (PHYD) model is widely adopted by the industry. In [11], the authors proposed a PHYD platform which has the ability of data acquisition, analysis, transmission and reasonable score computation. Without loss of the generality, in this study, we follow this PHYD model and introduce a general scoring method in Section III-B.

B. Location Privacy based on The Side-channel Inference

Location privacy is a long-standing topic [12], [13], [14], [15], [16]. Recently, there is an increasing interest in tracking the people's trajectory by leveraging the side-channel information including mobile device power [17], accelerometers [18] and other zero-permission mobile sensors [19]. Michalevsky *et al.* [17] proposed an approach which can use device power as the side-channel information to track users. The basic idea is the attacker can track a user by exploiting the correlation between the power consumption and the distance with the cellular tower. The previous works have discussed the location privacy in the new insurance mode, but the proposed algorithms suffer from limited tracking performance[7] [8].

Different from any previous work, we propose a novel approach to infer the trajectories by merely exploiting the driving speed, which outperforms the peer algorithms. What is more important, we present the first privacy-preserving data aggregation framework for UBI under the current architecture, which allows insurance companies to analyze the driver's driving habits without compromising his location privacy.

C. Secure Aggregation Protocol

The secure aggregation protocol is an approach to protecting the user's privacy while ensuring the computability of the uploaded data [20], [21], [22], [23], [24], [25]. A scheme that is based on cryptographic techniques and differential privacy was proposed in [20], which could be used to compute the aggregated sum on an untrusted server. Keith Bonawitz *et al.* [21] considered the situation when some users drop, and proposed a secure aggregation protocol that is based on one-time pads and secret sharing. Few works focus on the privacy-preserving architecture of the UBI. Carmela Troncoso *et al.* [26] presented PriPAYD, which locally performs the premium calculations and sends only aggregated data to the insurance company. However, only uploading aggregated data will require changes in the existing architecture that is used by the mainstream insurance companies. Moreover, the PriPAYD scheme can only resist GPS attacks (uploading full GPS data) which do not exist under the current insurance mode. Compared with PriPAYD, the proposed scheme can achieve the

privacy-preserving UBI without changing the current industry practice.

Furthermore, current secure aggregation protocols are not suitable our problem. In UBI, the insurer should choose specific data rather than all the data and sum them; current protocols do not achieve this goal. More important, current methods do not consider how to verify the authenticity of the uploaded data while ensuring their security. Based on this observation, we propose a novel privacy-preserving aggregation and data audition protocol that achieves our security and utility goals.

III. Overview of Usage-based insurance

UBI is a telematics-based insurance service in which premiums are based on driving behavior of consumers (e.g., braking and cornering pattern). Different from traditional insurance, in which premiums are based on driving history, in UBI, the insurance premium is calculated dynamically based not only on how much one drives but also on how and when one drives.

A. Driving Behavior Risk Indicators

Usage-based insurance relies on the following technologies to collect the data related to the driver's driving habits.

- *OBD-II*: Drivers plug a device into the vehicle's diagnostic port. It captures mileage, speed, braking and other measurements. Insurers often give them for free to prospective customers. This method is still expensive because of the device, data plan and distribution costs.
- *Smartphone*: In smartphone-based method, the installed application uses sensors to collect the metrics as OBD. A smartphone program like Drivewise can provide rating factors that are accurate enough for insurance premiums and cost $50-75\%$ less than an OBD program.
- *Hybrid OBD-Smartphone*: This approach combines smartphone and OBD based approaches.

We have surveyed the major players in US and China markets. Their interested data are summarized in Table I. According to Table.I, although the data collected by different companies are slightly different, some factors such as speed per second, hard braking, mileage, time of day are widely adopted by the insurers for insurance calculation. We will give a more detailed discussion on how to model the behavior-based insurance in the following.

B. Usage-based Insurance: Pay-How-You-Drive Model

Insurance policies that are based on vehicle use (usage-based insurance or UBI) include pay-as-you-drive (PAYD) and pay-how-you-drive (PHYD) systems. The PAYD system [6] charges premiums that are based on total travel behavior characteristics such as mileage and which roads network are used while in PHYD [27] premiums are based on parameters that measure individual driving behavior such as speed, harsh acceleration, and hard braking. Since evaluating how a user is driving is more critical for estimating the crash risk than how much he drives, the PHYD model is regarded as a more promising model for a UBI insurance policy. Therefore, before investigating the privacy threats, we briefly introduce the PHYD insurance model.

TABLE I: Data Collected from Some Mainstream Insurers

Company \ Data	Hard Braking	Mileage	Time of Day	Speed	Acceleration	Turn
StateFarm	√	√	√	√	√	√
Progressive	√	√	√	√		
Allstate	√	√	√	√		
Esurance	√		√	√	√	
ZhongHua Insurance	√	√		√	√	

The basic concept of PHYD insurance system is to construct a cost model based on how much (mileage), when (day/night) and how (overspeeding, harsh accelerations, hard brakes) a vehicle is driven [6]. Most of the existing researches [11] adopt a linear method to model PHYD insurance. Without loss of the generality, we consider the following PHYD model, which is a general model based on the existing industry practice and previous works.

$$\mathcal{P} = \mathcal{P}_b + \sum_{i=1}^{k} w_i * \mathcal{RS}_i, \tag{1}$$

where $\mathcal{P}$ refers to the total price of car insurance, $\mathcal{P}_b$ is the fixed charge, $\{\mathcal{RS}_i | 1 \leq i \leq k\}$ are the risk scores introduced by a particular driving behavior and $\{\omega_i | 1 \leq i \leq k\}$ are their corresponding weights. In the following, we take three potential behavior risks as an example. It is noted that it is easy to generalize this model to other behavior risks.

Risky Hour Driving: It is obvious that it is more dangerous to drive in the evening. So it is reasonable to assign a bigger risk factor α (> 0.5) to the driving at night due to the higher risk. The risk score of time of day $\mathcal{RS}_1$ can be written as:

$$\mathcal{RS}_1 = (1-\alpha) * time_{day} + \alpha * time_{night} \tag{2}$$

where $time_{day}$ and $time_{night}$ are the total travel time during the daytime and the evening, respectively.

Speeding: Speeding reflects the driver's driving habits and thus represents an important risk indicator. Without loss of generality, we set a threshold s_0, which is denoted as the speed limit (set as 80 mph in Allstate). Following the similar strategy in [11], we consider the speeds over the threshold and sum them up after subtracting the threshold, which is defined as:

$$\mathcal{RS}_2 = \textstyle\sum_{s_i > s_0} (s_i - s_0) \tag{3}$$

where $\mathcal{RS}_2$ is the score and s_i is the driver's speed per second.

Harsh Acceleration and Hard Braking: Hard braking and harsh acceleration are important indicators for risky driving. As pointed out by Allstate, hard braking events are recorded when the vehicle decelerates more than 8 mph in one second. We use the symbol of $C_{hd,acc}$ to denote the number of harsh acceleration/braking events.

Based on the discussion of three different kinds risk indicators, we get the final price by updating the Equation (1).

$$\begin{aligned}\mathcal{P} = \mathcal{P}_b + \omega_1 * [(1-\alpha) * time_{day} + \alpha * time_{night}] \\ + \textstyle\sum_{s_i > s_0} \omega_2 * (s_i - s_0) + \omega_3 * (C_{hb,acc})\end{aligned} \tag{4}$$

C. Privacy Concerns

Although some of the UBI are based on collected GPS information without considering any privacy issue, many other UBI programs are advertised as being privacy-preserving. Some insurance companies only record speed, mileage and hard braking, and claim that they do not intend to collect the user's location. According to existing research works [7] [8], it may be possible to infer driving routines based only on driving speed. However, the existing researches suffer from low successfully rates. In this paper, we will present a novel routine inference scheme, which can significantly improve the inference accuracy. Firstly, we will analyze the security and privacy issues of usage-based insurance and present the relative attack model and assumptions regarding the considered issues.

IV. Attack Model and Assumptions

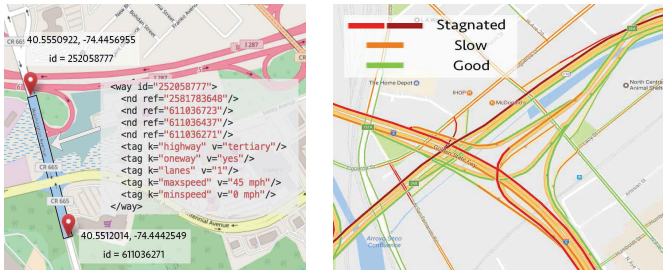
In this study, we jointly consider the security and privacy issues of UBI from the perspectives of insurance companies and drivers. On one hand, insurers are assume to be honest-but-curious, which means they are honest in executing the protocol, including data collection and score computation, but curious in inferring the driver's trajectories based on the collected data. On the other hand, since the data collection is performed at the user side, we consider that the misbehaving drivers may intentionally upload modified driving data to enjoy a lower premium. Existing researches [28], [29] show that it is technically possible for drivers to modify the GPS or sensor readings to mislead the insurer into thinking that he is "safe" and, thus, enjoy a lower premium. Therefore, it is reasonable to consider the data authentication problem from the perspective of insurers. Based on the above analysis, we define the following two attacks which will be considered in our paper:

- *Location Tracking Attack*: In this attack, the adversary may be the honest-but-curious insurer or the external attacker that can hack the storage system of insurance companies and gain access to the speed data of drivers. Similar to the attack model that was adopted in [7] [8], the attacker is assumed to have the initial location and speed data per second, and his aim is to track the target driver based only on these data.
- *Data Forgery Attack*: In this attack, a misbehaving or selfish driver may forge driving data by manipulating the GPS module or sensors so that he can mislead the insurer and enjoy a lower premium.

As shown in [28], attackers can generate fake data by manipulating the GPS module or sensors. Therefore, it is feasible and easy to launch the data forgery attack. In the next section, we mainly focus on how to execute the location tracking attack by presenting a novel routine inference system, which can significantly improve the inference accuracy.

V. Novel Trajectory Inference Attack System based on The Driving Speed

In this section, we introduce the location tracking attack in detail, which aims at inferring the driving trajectory from only the driving speed and the initial location. With the input of the initial location, speed data and pre-processed map information, the proposed inference system can automatically infer the driver's trajectory.



(a) Speed Limit From OSM (b) RRT Data From Google

Fig. 1: Real World Data

A. Attack Overview

The basic strategy of the proposed attack is based on the following insight: driving speed is influenced by many environmental factors, such as road condition, real-time traffic, and even traffic regulations. From the attacker's perspective, these environmental factors and the real-time speed provide the side-channel information about the driving routes, which can be exploited to filter out impossible routes and determine the most likely candidate routes. We summarize the environmental information as follows:

- *Advised Speed Limit for Non-Curved Roads:* Vehicles should follow advised speed limit s_{max} that is recommended by the government. If the vehicle exceeds the speed limit, the exceeding proportion must remain below a threshold σ. Otherwise, the driver will receive a speeding ticket. In general, advisory speed limit provides upper bound of the speed and can be automatically extracted from the map shown in Fig. 1(a). Therefore, the advisory speed limit can be exploited by an attacker to infer the user's trajectory. For further discussion, we give the definition as follow: for a series of speeds $S = \{s_1, s_2, \ldots, s_n\}$, the percentage of speeding is $\sum_{s_i} \mathbb{I}(s_i)/|S|$, where $\mathbb{I}(s_i) = 1$ if $s_i > s_{max}$, otherwise $\mathbb{I}(s_i) = 0$.
- *Impact of Real-time Road Traffic (RRT):* In practice, the speed of the vehicle is highly influenced by the road traffic, especially during rush hours. Many maps (e.g., Google maps and Baidu map) offer an API that can display real-time traffic for the road, which facilitates the tracking attack by the attacker. As shown in Fig. 1(b), different colors represent different traffic statuses, including: "Good" (e.g., driving at v_{max}), "Slow" (e.g., $\frac{2}{3}v_{max}$), and "Stagnated"(e.g., $\frac{1}{3}v_{max}$). Some map systems even offer the theoretical driving time t_{query} for a specific road segment based on the current road traffic.
- *Speed Limit for Driving through a Curve:* When a vehicle approaches a curve of radius r, the driver should slow down by following the speed limit. If the vehicle speed is higher than the speed limit, the vehicle speed control

system provides a warning [30]. Specifically, vehicle's friction must be greater than centrifugal force to ensure the safety:

$$f_r \cdot m \cdot g > m \cdot r \cdot (\frac{v_b}{r})^2 \quad (5)$$

where f_r is the friction coefficient of the road, m, g refer to the weight of the vehicle and the gravitational acceleration respectively. Then we can get the maximum turning speed $v_b = \sqrt{f_r \cdot g \cdot r}$. Note that, the parameters of the road can be obtained from the map.

Based on the above observations, it is possible for the external attacker to rule out less-likely routes and calculate the most-likely candidate routes. Our basic strategy which is illustrated in Fig. 2 is to build the speed model for various road conditions based on advisory speed limit, real-time road traffic, and speed limit for driving a curve. By comparing the speed model and the collected speed, we can calculate the probability for each road segment by adopting the Dynamic Time Warping (DTW) algorithm. The details of the proposed attack are presented in Section V-C.

B. Problem Formulation

It is assumed that the attacker knows the starting point of the trip and can exploit publicly available information (e.g., road information from OpenStreetMap) to launch the attack to track the target driver. Let N be the number of intersections in the area. A directed graph $G = (V, E)$ can be used to represent all the intersections. $V = \{1, 2, \cdots\}$ refers to the set of intersections and $e_{xy} \in E$ stands for a road between intersection x and intersection y. The basic goal of the attacker is to identify the route that fits the speed model best.

Then we formalize the problem of inferring users' trace as a Hidden Markov Model (HMM). We define the route of the vehicle as (Q, T):

$$Q = \{q_1, q_2, ...\}, q_i \in E \quad T = \{(t_0, t_1), (t_1, t_2), ...\} \quad (6)$$

where Q is the set of road segments and (t_{i-1}, t_i) are the start time and end time driving on the road q_i.

Formally, an HMM is characterized by the following:

- The state transition probability distribution $A = \{a_{e_{xy} \to e_{mn}} | \forall e_{xy}, e_{mn} \in E\}$, where

$$a_{e_{xy} \to e_{mn}} = p(q_{i+1} = e_{mn} | q_i = e_{xy}) \quad (7)$$

  $a_{e_{xy} \to e_{mn}}$ means the probability of directly moving from road e_{xy} to road e_{mn}. We set $a_{e_{xy} \to e_{mn}} = 0$ if $y \neq m$ and set other probabilities to be uniformly distributed over all possible transitions.
- The observation symbol probability distribution B. In our model, observation is a series of speed values and corresponding timestamps. We denote the observation as $O = \{o_1, o_2, \ldots\}$. Then B can be denoted as

$$B = \{b_{e_{xy}}(o_k) = p(o_k | q_t = e_{xy})\} \quad (8)$$

  where $b_{e_{xy}}(o_k)$ is the probability of generating the given speed values o_k while the vehicle drives through the road segment e_{xy}. In this work, this probability can be calculated by comparing the distance between the collected speed values and built speed model by running the DTW algorithm.
- We define $\pi = \{\pi_{e_{xy}}\}$ as the initial state distribution, where $\pi_{e_{xy}}$ is the probability that the vehicle initially goes through the road segment e_{xy}. Since the starting location (at a intersection) is known, we set obtain $\pi_{e_{on}}$ to $1/k$. Here, e_{on} refers to the neighboring road segments of original location and k refers to the total number of neighboring segments. Others is set to 0.

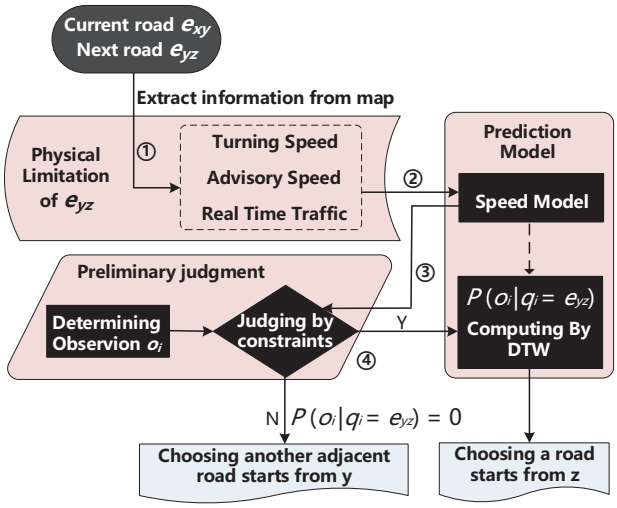


Fig. 2: The Framework of the Trajectory Inference Attack

We can define our problem as a classical HMM problem. Given an observation O, and $\lambda = (\pi, A, B)$ which denotes the parameter set of the model, our goal is to choose an optimal state sequence Q for this observation. According to [17], this problem is equivalent to maximizing $P(Q, O|\lambda) = P(O|Q, \lambda) \cdot P(Q|\lambda)$. Since the attacker has no prior knowledge about the drivers destination without the observation, we assume the driver has equal probability of travelling these routes. Therefore, this problem can be interpreted as the problem of finding an optimal route Q such that $P(O|Q, \lambda)$ is maximized. In the remainder of the paper, we denote $P(O|Q, \lambda)$ as $P(O|Q)$ for ease of presentation.

C. Attack Framework

For a route with m road segments, we split the observation into m sub-observations. Every pair of sub-observations o_i and o_j which are series of discrete speed values, are assumed to be mutually independent according to [31]. Therefore, based on the property of the output independence assumption, we can divide a route into many road segments and compute them iteratively:

$$\begin{aligned} p(O|Q) &= p(o_1, ..., o_m | q_1, \ldots, q_m) \\ &= \prod p(o_i | q_1, \ldots, q_m) \\ &= \prod p(o_i | q_i) \\ &= \prod b_{e_{xy}}(o_i) \end{aligned} \quad (9)$$

We can use the forward algorithm to calculate the probability of the state sequence in a specific HMM and find the

most possible sequence. However, the total number of possible routes in a large area may be too large for computation. In general, the complexity of this problem is $O(N^M)$, where N is the number of all roads and M is the possible number of road segments which vehicle traveled. So it is difficult to perform an exhaustive search in our problem. Fortunately, for the considered problem, we know the initial location and consider only four options for the next road segment, rather than randomly selecting from N roads. Thus, the complexity is reduced to $O(4^M)$. It is possible to further reduce the complexity by the following steps shown in Fig. 2:

- *Speed Model Checking*: In Section V-D, we introduce a speed model (Step. 2). At each iteration i, we will check whether o_i satisfies the constraints defined by the speed model (Step. 3). If it does not match, $p(o_i|q_i)$ is set to 0 and this road segment is not considered.
- *Probability Calculation*: If a road segment passes the speed model checking, which means it satisfies the speed limits and constraints. We can further calculate $p(o_i|q_i)$ by DTW (Step. 4), which represents the probability of the vehicle traversing a given segment q_i.

Algorithm 1: Pruning-based DFS(node,timestamp)

Input: original location x, currentTimestamp t_0
Output: candidate road segments H

$I_1 \leftarrow$ query_nextnodelist(x) by state transition matrix A;
for *every node* $y \in I_1$ **do**
  $d_i \leftarrow$ calculate_distance(e_{xy});
  $m_i \leftarrow$ generate_speed_model(e_{xy});
  $t_{start} \leftarrow$ currentTimestamp;
  $t_{end} \leftarrow$ determine_endtime(t_{start}, d_i);
  $o_i \leftarrow$ observation(t_{start}, t_{end});
  if $v_{turn} > v_b \ \ || \ \sum_{s_i} \mathbb{I}(s_i)/|S| \ > \sigma \ \ ||$
   $(t_{end} - t_{start}) \notin [\eta_1 \cdot t_{query}, \eta_2 \cdot t_{query}]$ **then**
    $p(o_i|q_i) = 0$;
    choose another node from I_1;
  end
  else
    $d_i = DTW(o_i, m_i)$;
    $d_s = DTW(m_i - m_{threshold}, m_i)$;
    $d_0 = DTW(0, m_i)$;
    get $p(o_i|q_i)$ by Equation 10
    $H \leftarrow$ store$(x, y, t_{start}, t_{end}, d_i, p(o_i|q_i))$;
    Pruning-based DFS (y, t_{end})
  end
  return H
end

Based on the above observations, we introduce a pruning-based depth-first search (DFS) algorithm for addressing the target problem. We introduce a pruning heuristic DFS so that candidate routes can be computed efficiently. The basic strategy is to remove the routes that contain road segments which are not reachable (the probability $p(o_i|q_i) = 0$) during the algorithm running time. Through setting reasonable constraints and speed model, we can remove most routes and generate the candidate routes from H. Then we sort the candidate routes according to $P(O|Q)$ to obtain the top-k routes. Algorithm 1 describes this strategy in pseudocode.

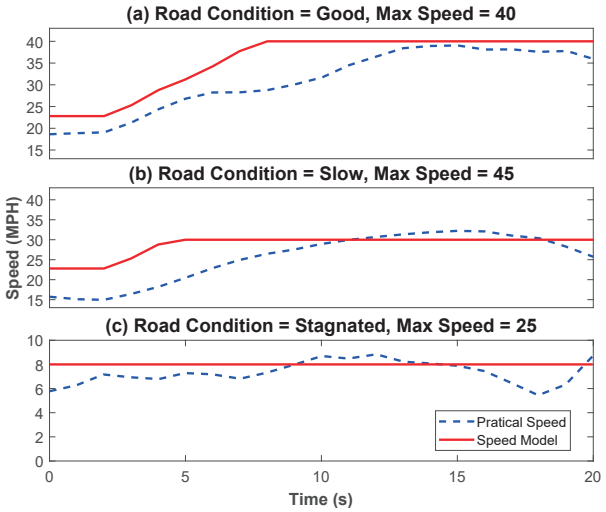


Fig. 3: Speed Model VS Practical Data

D. Building the Speed Model

The goal of the speed model is to filter out the impossible routes, which fail to satisfy the speed limits that were introduced in section V-A. The speed model describes a theoretical maximum speed for a road segment at a specific time and is defined by a series of continuous speed values. We use a method that was proposed in [7] and extend it by adding real time-traffic information to fit various situations. The basic model comes from the value of the maximum speed for each road, which can be collected from OpenStreetMap (OSM) and Wikipedia. If a turning event occurs, we improve the speed model by adding a turning speed limit which is calculated via the law of cosines according to the previous road segment.

Adding the real-time traffic information: It is obvious that the vehicle cannot reach its maximum speed limit of a specific road during the rush hour. Therefore, the real-time traffic is critical for improving the tracking precision. In practice, the real-time traffic has been provided by several navigation engines (e.g., Google map), which will be integrated into our speed model as shown in Fig. 3. This figure compares the practical data with the values produced from the speed model of a same road segment under different road conditions, which indicates that drivers' speed will be close to (maybe slightly greater than) the speed model. Based on the speed model, we can rule out the impossible routes and calculate the probability by DTW as shown in the following section.

E. Calculating the Probability of a Possible Route by DTW

Drivers are more likely to drive close to the maximum speed if road condition permits and safety is guaranteed. For example, drivers will not drive at a speed of 30km/h if the maximum speed is 60km/h. Thus, we have the following insight: the smaller the difference between the speed model and the collected data, the higher the probability of a specific route being the driving route of the target vehicle.

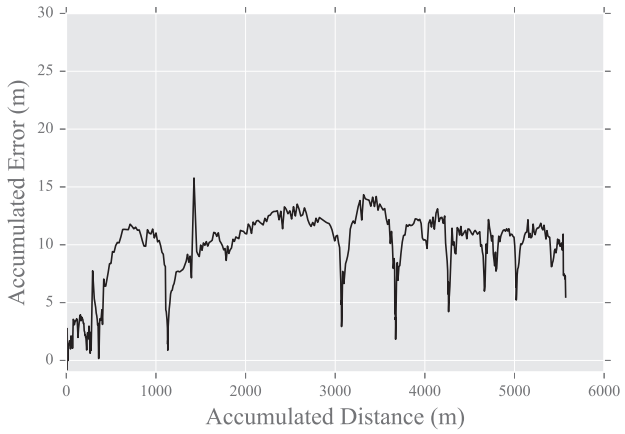


Fig. 4: Error between calculated distance and actual distance

DTW algorithm [32] is an algorithm that can compute the difference between two given sequences which may have different lengths (e.g. time series) with certain restrictions. We adopt the DTW algorithm to calculate the probability of the vehicle being on a specific route. In particular, given a specific road, the corresponding speed model and the observed speed are m_i and o_i, respectively. Then we compute the Euclidean distance d_i between the speed model and observation o_i, and the Euclidean distance d_0 between the speed model and zero. The latter indicates the maximum distance between the speed model and the possible vehicle speed. In addition, we define the Euclidean distance d_s between the speed model and the speed model minus a threshold, which means that the probability is the same if the driving speed is within the scope. Based on the above parameters, we can calculate the probability of a specific route being the driving route of the target vehicle as follows:

$$p(o_i|q_i) = \begin{cases} 1, & d_i \leq d_s \\ 1 - \frac{(d_i - d_s)}{(d_0 - d_s)}, & d_i > d_s \end{cases} \quad (10)$$

Based on the calculated probability, we can maintain a list of possible locations. Then we sort the candidate routes that are obtained from H and select the top-k routes.

F. Choosing the Corresponding o_i with Road Segment q_i

We know the time t_{start} when the vehicle enters into the road segment q_i. To obtain the corresponding o_i, we should determine the end time t_{end} leaving q_i. Because we only have pairs of speed per second and timestamp, we use a method for approximating the driving distance per second: we assume the movement at every second is a uniformly accelerated (retarded) rectilinear motion. Fig.4 shows the error between the calculated distance and the actual distance of a trip, which is small compared with the total distance.

According to the calculated distance per second and road distance, we can obtain the end time t_{end}. Then we can obtain the corresponding speed values in time interval $[t_{start}, t_{end}]$, which is o_i that corresponds to the segment q_i.

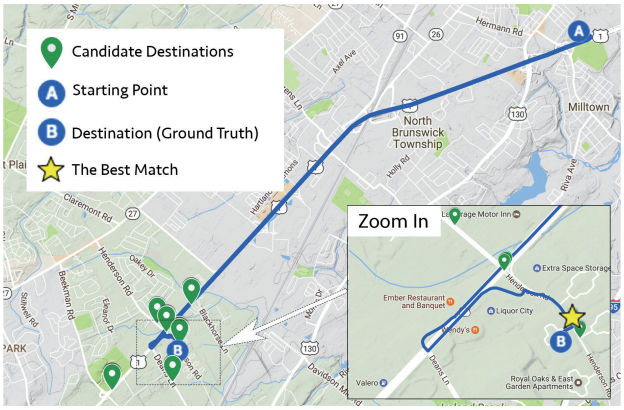


Fig. 5: Inferred Result of a Trip

G. Experimental Results

We launch our attack using data from 120 trips. Each trip varies in the range of $(7\ km, 21\ km)$ and the average length is $12\ km$. Experimental results show that the attacker has nearly 60% probability of obtaining the real route if he chooses the top 10 candidate routes of a trip. Fig. 5 shows the inferred candidate routes of a trip. The best match is an inferred route that completely matches the real route. Based on our algorithm, we can identify a few candidate routes (destinations of those routes are marked in green) from massive routes. Many of the candidate routes end are near the destinations. The main reason for this is that when a trip is nearly over, most users will slow down and speed limitations will not function properly. The detailed experimental results for all the trips are introduced in Section VII.

VI. Privacy-Preserving Scoring and Data Audition Framework

In this section, we will present our novel defense framework by jointly considering the location privacy attack and data forgery attack that are defined in the above sections, which can protect drivers' location privacy without compromising its utility.

A. Design Goals

As shown in the above sections, we know how to launch the two attacks: the location privacy attack and the data forgery attack. For thwarting the defined attacks, we design a privacy preserving scoring and data audition framework, which is denoted as Pri-UBI. The design goals of Pri-UBI are as follows:

- *Security Goal:* Pri-UBI is designed to defend against the location tracking attack and the data forgery attack. Our objective is to protect the driver's privacy while allowing the insurer to perform risk calculations.
- *Utility Goal:* With Pri-UBI, the insurer can still calculate the risk of a driver based on the collected data and verify the data's authenticity without leaking drivers' privacy.
- *Deployability Goal:* Pri-UBI is incrementally deployable as a complement to the existing system rather than displacing it, which should follow the current protocol.

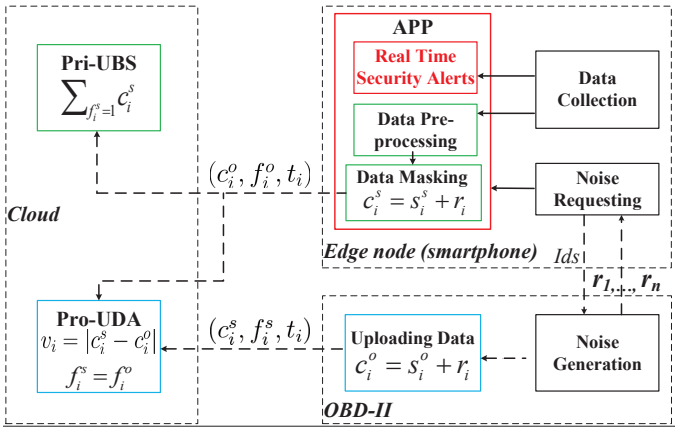


Fig. 6: Overview of our system

Our ultimate goal is to protect the driver's privacy and, at the same time, allow the insurer to perform risk calculation.

B. Framework overview

As shown in Fig.6, the proposed Pri-UBI is comprised of two parts: the Private Usage Based Scoring (Pri-UBS) protocol and Probabilistic Usage Data Audition (Pro-UDA) protocol, which are designed to enable insurers to calculate risk scores based on driving data without revealing their privacy and to prevent drivers from forging data. Our scheme serves as the interface between drivers and the insurer to provide privacy preserving functionality without changing the current system architecture.

To thwart the Location Tracking Attack, the Pri-UBS protocol use a novel privacy-preserving aggregation scheme to allow the insurer to perform the risk calculations without revealing individual data. In particular, Pri-UBS add noises (or random numbers) to the speed data that are indicated by a flag bit, which can be canceled after uploading.

To thwart the data forgery attack, the basic strategy of the Pro-UDA protocol is to verify the authenticity of driving data by checking the heterogeneous sensor readings from various sources (e.g., a smartphone and an OBD device that is plugged into the OBD-II port) [33], [34]. It is technically possible for users to manipulate the GPS readings or the OBD device [35]. However, existing research shows that the readings from different sensors on the OBD device or the smartphone are highly correlated. Therefore, we assume that malicious/selfish users cannot forge all the sensing data from smartphones or OBD devices, which allows the detection of the data forgery attack by cross-checking all the sensor-readings from various sources. However, this approach inevitably incur high communication and transmission overhead by transmitting all the sensing data to the cloud and introduces privacy concerns. Therefore, in this article, we propose a probabilistic checking framework that achieves data forgery attack detection with reduced overhead.

Since the insurance company is assumed to be honest-but-curious, the OBD device is expected to honestly follow the proposed encryption protocol (defined in this section). Although the OBD device is provided by the insurer, its misbehavior (e.g., Uploading) can be examined and detected by adopting the existing firmware analysis tools (e.g., Binwalk and IDA [36], [37]). Furthermore, along with General Data Protection Regulation (GDPR), a regulation in EU law on data protection and privacy for all individuals, coming into effect, the insurance companies are motivated to take actions to protect the customer's personal data (e.g., encryption) in compliance with the GDPR [38].

In addition to score computation, we refer to the smartphone as the edge node for providing real-time feedback reports on driving behaviors by analyzing these data, which can provide safe driving functions such as emergency response, speed alerts and real-time vehicle diagnostics.

C. Collected Data Format

In our survey, we examine the data that are collected by the mainstream insurance companies (e.g., Progressive, AllState, State Farm, and others) in Table.I. We present a symbolic representation of these data, which is defined as a five tuple:

$$Tr_i =< ML, T_a, S, H_c, A_c > \tag{11}$$

where Tr_i is the identifier of a trip, ML and T_a are the whole mileage and time of Tr_i respectively. S is a set, containing all pairs of speed data per second s_i and corresponding UNIX timestamp t_i, which can be shown as $S = \{(s_1, t_1), (s_2, t_2), ...(s_n, t_n)\}$. For further discussion, we use $S_s = \{(s_1^s, t_1), (s_2^s, t_2), \cdots, (s_n^s, t_n)\}$ and $S_o = \{(s_1^o, t_1), (s_2^o, t_2), \cdots, (s_n^o, t_n)\}$ to represent the speed data of a same trip collected by the smartphone and the OBD device respectively. We use H_c and A_c to represent the numbers of harsh braking and acceleration events.

According to the current UBI policy, insurer $\mathcal{I}$ aggregates the data that are collected from each driver to calculate the risk scores and determine the insurance premium as shown in Equation (4). Our proposed framework will ensure that insurer $\mathcal{I}$ can compute the speeding risk score $\mathcal{RS}_2 = \sum_{s_i > s_0}(s_i - s_0)$ securely. Specifically, the insurer $\mathcal{I}$ only knows the sum of the speed data that are greater than s_0 rather than any individual data.

D. Pri-UBS Protocol for Thwarting the Location Tracking Attack

In this section, we propose a novel Pri-UBS protocol for protecting drivers' privacy while allowing insurers to calculate the risk score based on driving data. The basic strategy of Pri-UBS is to add randomly generated numbers, which can be removed during the aggregation process [20]. Using this approach, the insurer can obtain the aggregated results without revealing any individual data. Different from any traditional approach, in UBI, insurer is more interested in data that directly impact the risk scores (e.g., the data that are greater than s_0, as defined in Equation (3)). Therefore, we further extend the privacy-preserving scheme by adding a series of flag bits that indicate sensitive data beyond a specified threshold.

1) Indicating Sensitive Data via Flag Bits: To enable the encryption of sensitive data, the driver should indicate whether the speed data s_i^s is greater than a specified threshold (e.g., the dangerous speed s_0), which can be achieved by adding a flag bit f_i^s for each data s_i^s, with a value of 0 for "no" and 1 for "yes". The flag bit enables the insurer to determine which data he needs to sum up without knowing their specific values. Note that, the user may modify the flag bits for cheating. We will address this issue in Section VI-E with the probabilistic audition scheme.

2) Privacy Preserving Speed Data Aggregation: To preserve the privacy of individual data, the proposed privacy preserving aggregation scheme introduces a randomly generated number before uploading. In particular, the smartphone sends the index set Id_s of the data with the flag bit $f_i^s = 1$ to the OBD device to obtain the random numbers, which are denoted as $Id_s = \{m_1, m_2, \ldots, m_k\}, m_j \in [1, n]$. After receiving the set Id_s, the OBD device generates k random numbers $\{r_{m_1}, r_{m_2}, \ldots, r_{m_k}\}$ via the following equation.

$$\sum_{j=1}^{k} r_{m_j} = r_{m_1} + r_{m_2} + \cdots + r_{m_k} = 0 \tag{12}$$

Then, it will generate another random number r_i with index $i \in \{1, 2, \ldots, n\} \backslash Id_s$ and send all of the random numbers $R_n = \{r_1, r_2, \ldots, r_n\}$ to the smartphone.

Note that, under the honest-but-curious model, the OBD device is expected to honestly follow the proposed encryption protocol and thus can be used to generate the random numbers.

3) Obfuscation with Random Numbers:

- **NoisyEnc**(param, r_i, s_i^s): At the smartphone side, speed data s_i^s is masked by the random number r_i as follows:

$$c_i^s = s_i^s + r_i \pmod R \tag{13}$$

  Under this situation, the speed data can be protected by the random numbers, which can ensure the computability.
- **AggrDec**(param, c_i^s, f_i^s, t_i): At the server side, after receiving the encrypted data $C^s = \{(c_1^s, f_1^s, t_1), (c_2^s, f_2^s, t_2), \ldots, (c_n^s, f_n^s, t_n)\}$, where t_i is the timestamp, the insurer selects the partial data whose flag bits are one and computes

$$\begin{aligned} \textstyle\sum_{f_i^s=1} c_i^s &= c_{m_1}^s + \cdots + c_{m_k}^s \\ &= s_{m_1}^s + r_{m_1} + \cdots + s_{m_k}^s + r_{m_k} \\ &= \textstyle\sum_{j=1}^{k} s_{m_j}^s + \sum_{j=1}^{k} r_{m_j} \\ &= \textstyle\sum_{j=1}^{k} s_{m_j}^s \pmod R \end{aligned} \tag{14}$$

  Since $\sum_{j=1}^{k} r_{m_j} = 0$, we can derive $\sum_{f_i^s=1} c_i^s = \sum_{j=1}^{k} s_{m_j}^s$, which means the sum of the speed data that exceeds the threshold s_0. Based on this result, insurers can calculate the speed risk score:

$$\mathcal{S}_2 = \sum_{s_i > s_0} (s_i - s_0) = \sum_{f_i^s=1} c_i^s - k \cdot s_0 \tag{15}$$

With the pri-UBS, the risk score can be calculated in a privacy-preserving manner without losing its accuracy. We'll discuss the privacy enhancement in the subsequent section.

4) Defending Against the Location Tracking Attack: In the Location Tracking Attack, insurer $\mathcal{I}$ constructs the relationship between the speed data and the roads based on the insight that the driving speed is influenced by many environmental factors, including real-time traffic and traffic regulations. However, it is impossible to determine the relationship between the encrypted speed and environmental factors because the encrypted data do not suit the physical limitations of the road (defined in Section V-A). $\mathcal{I}$ cannot identify the real route from massive routes. Thus, Pri-UBS can successfully thwart the Location Tracking Attack.

E. Pro-UDA protocol for Thwarting the Data Forgery Attack

In the previous section, we presented Pri-UBS, which is expected to protect the user's location privacy. In this section, we will discuss how to prevent malicious users from generating fake data to obtain a lower premium. We leverage the probabilistic audition to check whether drivers modify the data by comparing the speed data that are collected by the sensing devices (e.g., the smartphone and the OBD device) for the same trip.

1) Basic Flow of Pro-UDA: As described in Section VI-B, malicious/selfish users cannot forge all the sensing data from the smartphone or the OBD device. Based on the above insight, it is possible to verify the authenticity of the data by comparing these data from different devices. Therefore, we propose the following protocol: the insurer $\mathcal{I}$ audits the data (S_s and S_o) by requesting the smartphone and the OBD device to upload their encrypted data, and comparing these two data which have the same timestamp ($s_i^s \stackrel{?}{=} s_i^o$).

However, we need to consider the following issues:

(1) To ensure drivers' privacy, two pieces of data need to be encrypted. Therefore, how to keep the computability in the case of encryption? We solve this problem by encrypting the data that have the same timestamp with the same exchanged key (Section VI-D2) before uploading.

(2) Due to the measurement error, these two data may not be exactly equal and even have a significant difference. Fortunately, according to the existing study proposed in [39], it is possible to control the error in the range of $(0, 3.8)$ mph. Therefore, we should check whether the difference between these two data is in a reasonable range rather than just checking whether these two data are equal.

The detailed protocol is described as follows:

- **NoisyEnc**(param, r_i, s_i^o): To ensure the computability for auditing, s_i^o is encrypted with the same random number r_i generated in Section VI-D2:

$$c_i^o = s_i^o + r_i \pmod R \tag{16}$$

  Then the OBD device sends the encrypted data $C^o = \{(c_1^o, f_1^o, t_1), (c_2^o, f_2^o, t_2), \ldots, (c_n^o, f_n^o, t_n)\}$ to the insurer $\mathcal{I}$, where f_i^o denotes the flag bit.
- **Data Audition**(param, s_i^s, s_i^o, t_i): After receiving the encrypted data, the insurer could get the difference v_i of these data whose timestamps are equal:

$$v_i = |c_i^s - c_i^o| = |s_i^s - s_i^o| \tag{17}$$

TABLE II: Payoff Matrix

M_0 \ M_1	B_c	B_n
A_m	$-F_s - C_u, F_s - I_c$	$R_w + L_d, -R_w$
A_n	$R_w - C_u, R_i - R_w - I_c$	$R_w, R_i - R_w$

With the proposed approach, the insurer can detect whether a driver is honest by comparing v_i with a threshold v_0,to determine whether the driver modifies his data. To alleviate the issue that are caused by measurement error, drivers will be punished only when mismatched data appear multiple times. Then, the insurer will impose a punishment F_s or reward R_w which can influence insurance premium based on the results.

To ensure the authenticity of the flag bit, the insurer should compare the corresponding flag bit for every pair of data ($f_i^s \stackrel{?}{=} f_i^o$) in C^s and C^o and set the count C_f as $C_f = \sum_{f_i^s < f_i^o}(f_i^o - f_i^s)$, which means the insurer only counts when $f_i^s = 0$ and $f_i^o = 1$. If C_f is greater than a threshold, the insurer $\mathcal{I}$ regards the driver as a dishonest user.

2) Leveraging Probabilistic Audition to Reduce the Overhead: In general, most drivers are honest. Therefore, it is unnecessary to audit all the data since this will incur substantial overhead. To reduce the cost that is incurred by auditing, we refer to a probabilistic misbehavior detection scheme [40]. First, we model the driver's and the insurer's actions as an inspection game. Then, we demonstrate that we can achieve audition with minimum cost by setting an appropriate auditing probability. The game is defined as $G =< M, \{e_0, e_1\}, \{\pi_0, \pi_1\} >$:

- $M = \{M_0, M_1\}$ is the set of players. M_0 denotes the driver, and M_1 denotes the insurer.
- e_0 and e_1 are the sets of players' strategies. Drivers have two strategies, modifying (A_m) and not modifying (A_n). Insurers also have two strategies, auditing (B_c) and not auditing (B_n).
- π_0, π_1 are the payoffs of players.

Then we define the cost and payoff of the players as follows:

- F_s is the punishment if the driver is detected for modifying and R_w is the reward if the driver is honest.
- We use C_u and I_c denote the cost that the driver spends to upload the data and the cost to check the data by insurers.
- L_d is the benefit of the driver if he modifies the data. Similarly, the insurer has the benefit R_i if the driver honestly uploads the data.

We assume p_a is the modifying probability, and p_b is the auditing probability. Then we can obtain the payoff matrix of the players as shown in Table. II.

We can compute a Nash Equilibrium point by the equation:

$$\begin{aligned}(-F_s - C_u)\cdot p_b + (R_w + L_d)\cdot(1-p_b) &= (R_w - C_u)\cdot p_b + R_w\cdot(1-p_b)\\ (F_s - I_c)\cdot p_a + (R_i - R_w - I_c)\cdot(1-p_a) &= -R_w\cdot p_a + (R_i - R_w)\cdot(1-p_a)\end{aligned} \tag{18}$$

where

$$(p_a^*, p_b^*) = (\frac{I_c}{F_s + R_w}, \frac{L_d}{F_s + R_w + L_d}) \tag{19}$$

It means the driver has the same payoff regardless of the choice of the strategy if insurer the audits data at the probability p_b^*.

Then we set the auditing probability $p_b = \frac{L_d+\delta}{F_s+R_w+L_d}, \delta > 0$, and compute the driver's payoff of the two strategies:

$$\begin{aligned}\pi_0(A_n) - \pi_0(A_m) &= (F_s + R_w + L_d)\cdot\tfrac{L_d+\delta}{F_s+R_w+L_d} - L_d\\ &= \delta > 0\end{aligned} \tag{20}$$

It means the driver has more payoff if he chooses the strategy of "A_n" when $p_b > \frac{L_d}{F_s+R_w+L_d}$, so rational driver will choose this strategy. The insurer can set a higher punishment to reduce the audition probability while ensure the detection accuracy.

3) Defending Against the Data Forgery Attack: In the Pro-UDA protocol, if a driver modifies the speed data or the flag bit, this dishonest behavior will be detected by the detection scheme. Inspection game can ensure that if the auditing probability is greater than $\frac{L_d}{F_s+R_w+L_d}$, a rational user will follow the protocol. Furthermore, the probability that a dishonest driver cannot be detected after m rounds is $(1 - \frac{L_d+\delta}{F_s+R_w+L_d})^m \to 0$, if $m \to \infty$. Thus, the security of the protocol can be ensured.

Since the OBD device obeys the protocol and does not collude with $\mathcal{I}$ (as described in Section VI-B), insurer $\mathcal{I}$ has no idea about the speed data. Thus, the Pro-UDA protocol can thwart the Data Forgery Attack defined in Section VI-A while ensuring drivers' privacy. By combining the pri-UBS and pro-UDA protocols, we can precisely compute the real risk score without compromising the driver's location privacy.

F. Security Analysis

In this section, we give a formal proof that insurer $\mathcal{I}$ can know only the sum of partial data for a driver. First, we define the following security game to describe our security model. Then, we refer to the proof that was proposed in [20], [21] to provide the security proof of our scheme.

Setup In this phase, the challenger generates random numbers $r_1, r_2, \ldots, r_n \in \mathbb{Z}_p$ such that $\sum_{j=1}^{k} r_{m_j} = 0$, where r_{m_j} is the index whose flag bit is equal to one.

Query The adversary chooses the index from the set $U \subseteq \{1, 2, \ldots, n\}$ and requests the ciphertext of these indices.

Challenge phase In the challenge phase, the challenger randomly flips a coin m. The challenger sends the real encrypted results $\{c_i | i \in U\}$ if $m = 0$. Otherwise, he randomly chooses n elements $c_1^{'}, c_2^{'}, \ldots, c_n^{'}$ which meets

$$\sum_{i\in U} c_i^{'} = \sum_{i\in U} c_i \tag{21}$$

Guess The adversary outputs a guess of whether m is 0 or 1. We say that the adversary wins the game if he correctly guesses m.

Theorem 1. *Our scheme is computational security if no probabilistic-polynomial adversary has more than non-negligible advantage in winning the above security game.*

Proof. In the following, we will prove the security of our scheme by the hybrid argument. First, we introduce the following n hybrid games. In $Game_d$, we define the information that the challenger sends to the adversary as

$$R_1^{'}, R_2^{'}, \ldots, R_d^{'}, c_{d+1}, \ldots, c_n \tag{22}$$

where $R_i^{'}$ is a random number that has the following property:

$$\sum_{1\le i\le d} R_i^{'} = \sum_{1\le i\le d} c_i \tag{23}$$

If $d = 0$, the challenger sends $c_1, c_2, \ldots, c_n$, which corresponds to the case of $m = 0$. Similarly, $Game_n$ corresponds to the case of $m = 1$.

To prove the security of our scheme, we should demonstrate that neighboring games $Game_{d-1}$ and $Game_d$ are computationally indistinguishable, which means attackers do not have a non-negligible advantage in differentiating the random number and the ciphertext.

In the following, we will compute the correlation between the predictive value and the ciphertext based on the concept of mutual information, which is the information about the plaintext that the attacker could obtain from the ciphertext.

Definition 1. *(Mutual Information) Let c_i denote the ciphertext of the speed data and $\tilde{s_i}$ denote the predictive value of the speed data by the attacker. We define the mutual information $-I(\tilde{s_i}, c_i)$ as:*

$$-I(\tilde{s_i}, c_i) = -\sum_{\tilde{s_i}=0}^{s_{max}} \sum_{c_i=0}^{s_{max}} p(\tilde{s_i}, c_i) \log \frac{p(\tilde{s_i}, c_i)}{p(\tilde{s_i})p(c_i)} \tag{24}$$

We assume the encryption has the same plaintext and ciphertext spaces, and s_{max} is the maximum value of the $\tilde{s_i}$ and c_i.

The equation indicates the possibility of the attacker to infer the real value from the ciphertext, which is the successful guess rate of the attacker. According to the definition, mutual information is non-positive, and this metric is inversely proportional to the information that the attacker learns from the ciphertext.

According to the law of total probability, we have,

$$p(c_i) = \sum_{k=0}^{s_{max}} p(c_i|\tilde{s_i} = k)p(\tilde{s_i} = k) \tag{25}$$

$p(\tilde{s_i}, c_i)$ is the joint probability distribution. We also have,

$$p(\tilde{s_i}, c_i) = p(c_i|\tilde{s_i})p(\tilde{s_i}) \tag{26}$$

Suppose α_j is the rate that the encryption scheme chooses the random number r_j, we have

$$p(c_i|\tilde{s_i}) = \sum_{j=0}^{s_{max}} \alpha_j [p(s_i \Leftrightarrow c_i|\tilde{s_i}) + p(s_i \not\Leftrightarrow c_i|\tilde{s_i})] \tag{27}$$

where $\Leftrightarrow$ represents c_i is the ciphertext of s_i, and $\not\Leftrightarrow$ has the opposite meaning. Especially, our scheme choose the random number independently, which can ensure $\forall j$, $\alpha_j = \frac{1}{|s_{max}|}$.

If the attacker has no prior knowledge about the plaintext and there is no collusion attack, we could find the probability $p(\tilde{s_i})$ fits a uniform distribution and $p(\tilde{s_i} = k) = \frac{1}{|s_{max}|}$. Then we have,

$$-I(\tilde{s_i}, c_i) = 0 \tag{28}$$

Based on the above analysis, the mutual information is zero, which means the attacker cannot obtain any information about the plaintext from the ciphertext. The mutual information between the predictive value $\tilde{s_i}$ and the random number $R^{'}$ is also zero. Therefore, the adversary can obtain the same information from the random number and the ciphertext, which means he/she cannot differentiate them and neighboring games $Game_{d-1}$ and $Game_d$ are computationally indistinguishable. □

G. The Transmission Cost of Pri-UBI

We evaluate the transmission cost of the proposed scheme by the following theorem.

Theorem 2. *Given n as the cardinal number of the set S and m as the number of bits of the random number, the transmission cost is at most $3mn$ bits.*

Proof. In the key exchanging phase, the OBD device will send the random numbers R_n to the smartphone for encryption. Therefore, it should send at most mn bits. The cost of sending the index by the smartphone can be ignored.

In the uploading phase, the smartphone sends the encrypted data and the OBD device probabilistically uploads the data. Thus, the total size of the data is less than $2mn$ bits. □

By using an inspection game, it is not necessary to collect the data and send it to the insurer. The OBD device needs to only collect, encrypt and upload the data at the probability $p_b = \frac{L_d+\delta}{F_s+R_w+L_d}$. under other conditions, It is used as a random number generator. Therefore, we can obtain a lower transmission cost from the probabilistic audition.

VII. Implementation and Evaluation

In this section, we first implement our attack framework on a real world dataset to demonstrate the performance in terms of the inference accuracy. Then, we evaluate the efficiency of the proposed Pri-UBI under different parameters with respect to the audition probability and data size.

A. Implementation of the Inference System

We use a public dataset [8] that was collected by volunteers in New Jersey, which contains the timestamped speed data and the ground truth of GPS data. We fetch the street information from OSM, including nodes, ways and relations. In addition, given that most cars travel without speeding, the value of the maximum speed for each path is a vital criterion in our estimation. Some of the values could be obtained from OSM, while others are from Wikipedia and the Department of Motor Vehicle (DMV) in United State.

As the real-time traffic is taken into consideration in our algorithm, we implement this part by calling Google Maps API. For Google Map only provides traffic conditions at present or in the future, we could not fetch it corresponding to the collection time of the dataset. However, the real-time traffic have a close relationship with historical data and often fluctuate within a small scope, so it is reasonable to request the traffic in the same time slot of the day as the traces.

In our experiments, we set $\sigma = 0.2$, $\eta_1 = 0.5$ and $\eta_2 = 2$. With the input of the initial location, speed data and pre-processed map information, as described above, our

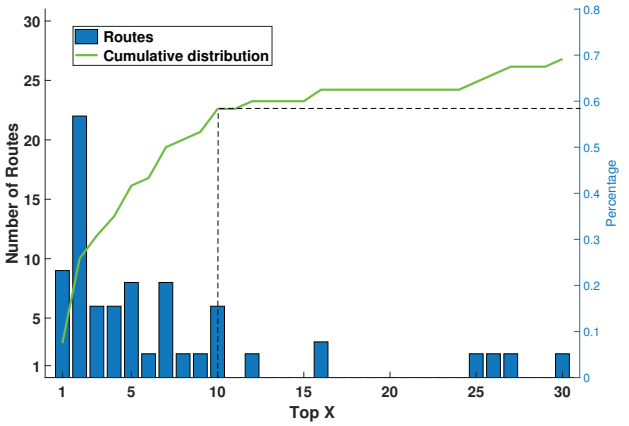


Fig. 7: The Accuracy of Our Inference Framework

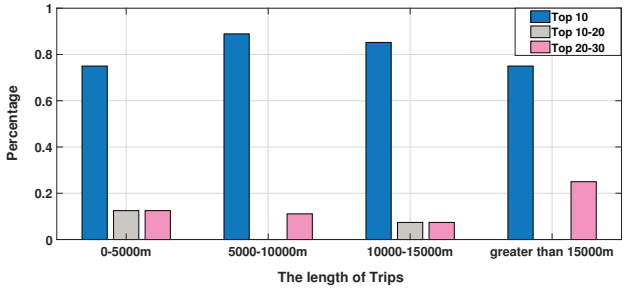


Fig. 8: The Relationship between the Distance and the Inference Accuracy

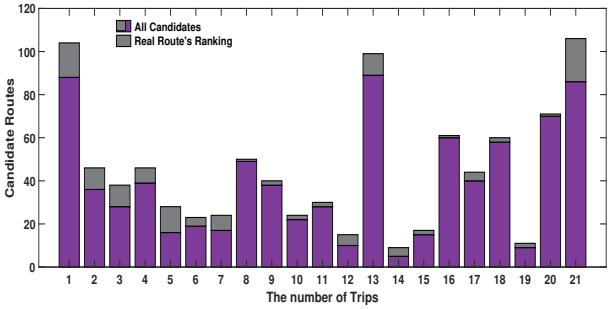


Fig. 9: The Ranking of The Real Route

trajectory inference framework can automatically infer the driver's candidate routes. For convenience of description, we define an inferred route as *the best match* if the inferred route completely matches the real route.

1) Route distinguishability: To evaluate the framework for distinguishing the real route from massive routes, we launch our attack with the data from 120 trips, which vary in the range of $(7\ km, 21\ km)$ and the average length is approximately $12\ km$. After generating the candidate routes, we will sort them according to the probability of each route, which is computed by DTW. This probability describes the degree of similarity between a candidate route and the real route.

In Fig. 7, the histogram shows the ranking (probability) of the real route among the candidate routes. For example, top one means the real route has the highest probability among massive candidate routes. According to this histogram, the ranking of most trips' real route are always in the top 10. The line chart shows the accuracy of our inference framework for selecting the real route among the candidate routes. There is a probability of nearly **60**% when selecting the top 10 candidate routes, which means the attacker has nearly 60% probability of obtaining the real route if he chooses the top 10 candidate routes of a trip.

Due to the influence of positively reducing the speed at the end of a trip, different driving habits, and the high updating frequency of ground information, it is difficult to accurately select the real route from massive routes in practice. Therefore, we introduce a new metric, top 10 routes, to show the performance of our inference system. In practice, choosing top 10 routes is sufficient for attackers to obtain extra information for inferring drivers' location privacy [41], [42], which can be used to narrow down the candidate routes (approximately 10^{56} for a $12\ km$ trip [8]) as much as possible.

Then, we evaluate the relationship between the trip's length and the inference accuracy in Fig. 8. Our evaluation shows that the accuracy does not decrease with increasing trip lengths, which demonstrates the stability of our algorithm under various situations. Furthermore, we list the candidate routes and real route's rankings (the junction of the two colors, the ranking is larger from top to bottom) of the trip whose length exceeds 10000m in Fig. 9. It can be seen from the sorting results that the real route's ranking is always on the top of all candidates regardless of the number of candidates, which means DTW performs well in selecting the best match (real route) from the candidate routes.

To illustrate the function of the real time traffic, we conduct a set of comparative experiments. Table III shows that real-time traffic has a significant effect on reducing the recursive times and increasing the inference accuracy.

2) Endpoint Error between Candidate Routes and the Real Route: To further show the performance of our trajectory inference system, we define another metric for calculating the endpoint error between candidate routes and the real route in Table IV. First, we compare the average error, which is defined as the average distance between the destinations of the real route and the candidates (we only compute the candidates whose rankings is greater than the real route) destination. The results show the percentage of the trips is 35.30% when the average endpoint error is within 0.5 km and 67.65% when the average endpoint error is within 1 km.

3) Performance Comparison with Other Method: To evaluate the proposed scheme's performance, we compare our

TABLE III: Effect of RRT

Without RRT		With RRT	
rank/candidates	recursion times	rank/candidates	recursion times
4/23	719	3/12	664
3/24	545	2/22	501
16/70	460	8/31	309
2/40	315	1/30	154
5/16	566	4/11	452
9/33	2732	5/15	2195
3/32	358	3/22	220
4/9	507	4/6	286
26/136	877	20/106	675

TABLE IV: Endpoint Error Between Candidate Routes and the Real Route

Endpoint error (avg.)		Endpoint error (Top One)		
Error (m)	Percent	Error (m)	Percent (Our Scheme)	Percent (Ubicomp' 14)
0-500	35.30%	0-500	35.29%	23.62%
500-1000	32.35%	500-1000	41.18%	9.84%
>1000	32.35%	>1000	23.53%	66.54%

TABLE V: Costs for OBD to generate random numbers

Length/bits \ Amount	1800	3600	5400	7200	9000
512	11ms	21ms	30ms	42ms	52ms
1024	20ms	41ms	62ms	81ms	102ms
2048	41ms	81ms	122ms	162ms	205ms

scheme with the method proposed in Ubicomp' 14 [8], which is the most closely related work to ours. This method (Ubicomp' 14) was able to identify an endpoint for a trip, and used the endpoint error which represents the distance between the the real route's destination and the endpoint to evaluate its performance. We select the candidate route whose probability is the highest (the method that was proposed in Ubicomp' 14 only inferred a candidate route for a trip) and compute the endpoint error with the real route's destination. According to Table IV, 76.47% $(35.29\% + 41.18\%)$ of trips have an endpoint error that is within 1 km, which is far higher than the percentage that is claimed in Ubicomp' 14 (33.46%). Our experiment demonstrates that our scheme has smaller tracking error compared with the other method.

B. Encryption Ratio's Impact on the Security

Encryption will generate additional overhead, which may affect the performance of the system. Therefore, it is reasonable to adjust the encryption ratio to seek a balance between the security and the performance. To evaluate the encryption ratio's impact on the security, we define a new metric, $\delta\%$ Pri-UBI, which means we randomly encrypt $\delta\%$ of the data to reduce the overhead. Then, we evaluate the inference success rate under different δ. Since the attacker cannot launch the location tracking attack with the ciphertext, we consider a smarter attack: the attacker replaces the ciphertext based on the adjacent plaintext (speed data).

In the experiment, we set parameter $\delta = 50$ or 100. From the Fig. 10, we could see that the attacker cannot infer any route after deploying the 100% Pri-UBI (red line), which demonstrates the effectiveness of the proposed scheme. With the 100% Pri-UBI, it is impossible to find the relationship between the encrypted speed data and environmental factors because the encrypted data do not fit the physical limitations of the road (defined in Section V-A). Thus, the attacker cannot infer any useful information. With 50% Pri-UBI (the middle line), the attacker could infer drivers' routes with a very low probability (12.8% when choosing top 10 routes). Therefore, we could adjust the encryption ratio to reduce the overhead according to the privacy requirements.

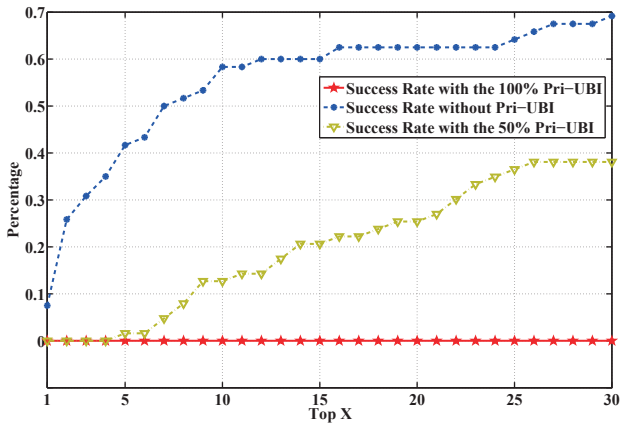


Fig. 10: Inference success rate with and without Pri-UBI

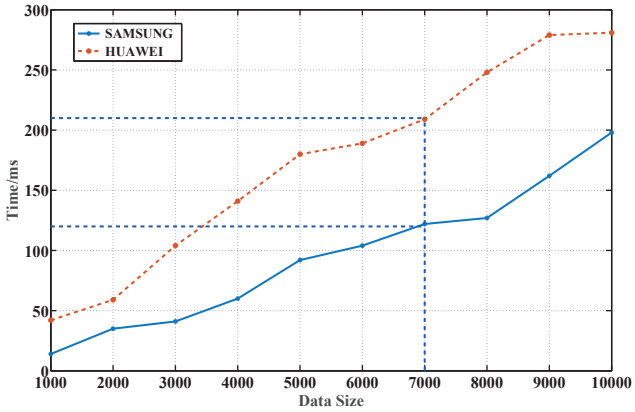


Fig. 11: Cost of data encryption on smartphone

C. Efficiency of Pri-UBI

The proposed protocol is evaluated on three types of devices: a computer, smartphones and a microcontroller that has similar calculation ability to the OBD device. We use a computer with an Intel i5 CPU of 2.8 GHz, two android smartphones with a Exynos 2.1 GHz CPU and a Snapdragon 1.5 GHz CPU respectively, as well as a 32-bits microcontroller with Intel Atom CPU of 500 MHz as the implementation platform. We evaluate the efficiency of Pri-UBI under different parameters with respect to audition probability and data size.

1) Cost of Random Number Generation at Microcontroller Side: The first metric measures the ability of microcontrollers to generate random numbers. We represent the cost of generating random numbers by the computation latency. The relation between the data size and the bits of random number is shown in Table. V. We can see the latency has increased from $20ms$ to $102ms$ with the data size changing from 1800 to 9000 when random number is 1024 bits. It indicates the microcontroller can generate random number easily.

2) Cost of Data Encryption at User Side: The second metric is about encrypting data at user side. We set random number as 1024 bits and evaluate it by different smart phones. Fig. 11 shows that the cost of encryption is linear with the data size. Also, the execution cost of the user side is influenced by the data size directly. Generally, the cost on user side can be completed within $300ms$.

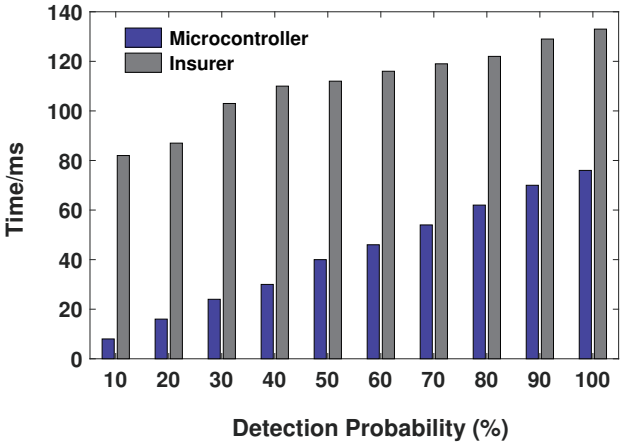


Fig. 12: Cost of different detection probabilities

3) Impact of Choosing Different Detection Probability: In this section, we evaluate the impact of the different detection probability on the performances of the microcontroller and the server. We set random number as 1024 bits and the driving time as one hour, which will generate 3600 data instances. Fig.12 illustrates that our protocol performs well in reducing the cost. The transmission cost can therefore be cut off by reducing the detection probability. And insurance companies could set a proper punishment to ensure the lower detection probability, which can reduce the cost of both sides.

VIII. Conclusion

In this paper, we find that attackers can track drivers by only the speed data and their initial location. By using the physical limitations of a road, attackers can identify the possible routes from the massive routes. To thwart the attack, we propose Pri-UBI, which can protect the user's privacy while not affecting score computation. Our scheme can also detect users' dishonest behavior via a probabilistic audition. We demonstrate the efficiency of our discovered attack and the proposed framework in the detailed experiments.

Acknowledgment

This work is supported by National Science Foundation of China (No. U1405251, U1401253, 61672350), and JSPS KAKENHI Grant Number JP16K00117, JP15K15976, KDDI Foundation.

References

[1] L. Zhou, Q. Chen, Z. Luo, H. Zhu, and C. Chen, "Speed-based location tracking in usage-based automotive insurance," in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, 2017, pp. 2252–2257.

[2] Progressive insurance company. [Online]. Available: https://www.progressive.com/auto/discounts/snapshot/

[3] Allstate insurance company. [Online]. Available: https://www.allstate.com/drive-wise.aspx

[4] State farm mutual automobile insurance company. [Online]. Available: https://www.statefarm.com/customer-care/download-mobile-apps/drive-safe-and-save-mobile

[5] The travelers companies inc. [Online]. Available: https://www.travelers.com/car-insurance/programs/Intellidrive

[6] D. I. Tselentis, G. Yannis, and E. I. Vlahogianni, "Innovative motor insurance schemes: a review of current practices and emerging challenges," *Accident Analysis & Prevention*, vol. 98, pp. 139–148, 2017.

[7] R. Dewri, P. Annadata, W. Eltarjaman, and R. Thurimella, "Inferring trip destinations from driving habits data," in *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. ACM, 2013, pp. 267–272.

[8] X. Gao, B. Firner, S. Sugrim, V. Kaiser-Pendergrast, Y. Yang, and J. Lindqvist, "Elastic pathing: Your speed is enough to track you," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 975–986.

[9] P. Händel, J. Ohlsson, M. Ohlsson, I. Skog, and E. Nygren, "Smartphone-based measurement systems for road vehicle traffic monitoring and usage-based insurance," *IEEE Systems Journal*, vol. 8, no. 4, pp. 1238–1248, 2014.

[10] Y. Bian, C. Yang, J. L. Zhao, and L. Liang, "Good drivers pay less: A study of usage-based vehicle insurance models," *Transportation Research Part A: Policy and Practice*, vol. 107, pp. 20–34, 2018.

[11] L. Boquete, J. M. Rodríguez-Ascariz, R. Barea, J. Cantos, J. M. Miguel-Jiménez, and S. Ortega, "Data acquisition, analysis and transmission platform for a pay-as-you-drive system," *Sensors*, vol. 10, no. 6, pp. 5395–5408, 2010.

[12] Y. Zhang, Y. Mao, and S. Zhong, "Joint differentially private gale–shapley mechanisms for location privacy protection in mobile traffic offloading systems," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 10, pp. 2738–2749, 2016.

[13] C. Chen, X. Zhu, P. Shen, J. Yu, H. Zou, and J. Hu, "Sols: A scheme for outsourced location based service," *Journal of Network and Computer Applications*, vol. 56, pp. 158–165, 2015.

[14] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016, pp. 546–563.

[15] Y. Li, L. Zhou, H. Zhu, and L. Sun, "Privacy-preserving location proof for securing large-scale database-driven cognitive radio networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 563–571, 2016.

[16] P. Zhou, W. Wei, K. Bian, D. O. Wu, Y. Hu, and Q. Wang, "Private and truthful aggregative game for large-scale spectrum sharing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 463–477, 2017.

[17] Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh, and G. Nakibly, "Powerspy: Location tracking using mobile device power analysis," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 785–800.

[18] J. Hua, Z. Shen, and S. Zhong, "We can track you if you take the metro: Tracking metro riders using accelerometers on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 286–297, 2017.

[19] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, "Inferring user routes and locations using zero-permission mobile sensors," in *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016, pp. 397–413.

[20] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Annual Network & Distributed System Security Symposium (NDSS)*. Internet Society., 2011.

[21] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017.

[22] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara, "High-throughput semi-honest secure three-party computation with an honest majority," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 805–817.

[23] Y. Zhang, Q. Chen, and S. Zhong, "Privacy-preserving data aggregation in mobile phone sensing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 980–992, 2016.

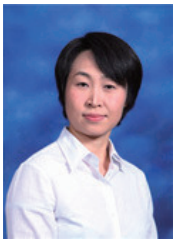
[24] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[25] Q. Wang, J. Wang, S. Hu, Q. Zou, and K. Ren, "Sechog: Privacy-preserving outsourcing computation of histogram of oriented gradients in the cloud," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 2016, pp. 257–268.

[26] C. Troncoso, G. Danezis, E. Kosta, J. Balasch, and B. Preneel, "Pripayd: Privacy-friendly pay-as-you-drive insurance," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 5, pp. 742–755, 2011.

[27] P. Desyllas and M. Sako, "Profiting from business model innovation: Evidence from pay-as-you-drive auto insurance," *Research Policy*, vol. 42, no. 1, pp. 101–116, 2013.
[28] A. Ranganathan, H. Ólafsdóttir, and S. Capkun, "Spree: a spoofing resistant gps receiver," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 348–360.
[29] M. Xue, C. Ballard, K. Liu, C. Nemelka, Y. Wu, K. Ross, and H. Qian, "You can yak but you can't hide: Localizing anonymous social network users," in *Proceedings of the 2016 ACM on Internet Measurement Conference*. ACM, 2016, pp. 25–31.
[30] S. C. Harris, "Non real time traffic system for a navigator," May 5 2015, uS Patent 9,026,358.
[31] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
[32] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.
[33] M. Amoozadeh, A. Raghuramu, C.-N. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, 2015.
[34] A. Ganesan, J. Rao, and K. Shin, "Exploiting consistency among heterogeneous sensors for vehicle anomaly detection," SAE Technical Paper, Tech. Rep., 2017.
[35] M. Li, H. Zhu, Z. Gao, S. Chen, L. Yu, S. Hu, and K. Ren, "All your location are belong to us: Breaking mobile social networks for automated user location tracking," in *Proceedings of the 15th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2014, pp. 43–52.
[36] A. Luo, "Drones hijacking," 2016.
[37] R. Santamarta, "Here be backdoors: A journey into the secrets of industrial firmware," *Black Hat USA*, 2012.
[38] (2018) Eu general data protection regulation. [Online]. Available: https://en.wikipedia.org/wiki/General_Data_Protection_Regulation
[39] A. Chowdhury, T. Chakravarty, and P. Balamuralidhar, "A novel approach to improve vehicle speed estimation using smartphones ins/gps sensors," in *IEEE International Conference on Sensing Technology (ICST), Liverpool, UK*, 2014, pp. 441–446.
[40] H. Zhu, S. Du, Z. Gao, M. Dong, and Z. Cao, "A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 22–32, 2014.
[41] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When csi meets public wifi: Inferring your mobile phone password via wifi signals," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1068–1079.
[42] J. Bellatti, A. Brunner, J. Lewis, P. Annadata, W. Eltarjaman, R. Dewri, and R. Thurimella, "Driving habits data: Location privacy implications and solutions," *IEEE Security & Privacy*, vol. 15, no. 1, pp. 12–20, 2017.

Lu Zhou received the B.Eng degree in Computer Science and Technology from Sichuan University, China, in 2015. He is currently pursuing the Ph.D. degree in Computer Science and Technology at Shanghai Jiao Tong University, China. His main fields of research interest include security and privacy in Vehicular network and Cognitive Radio Networks.

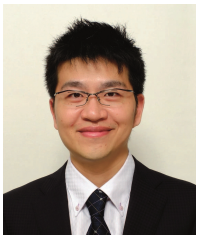


Suguo Du is an Associate Professor in Department of Management Science, Shanghai Jiao Tong University, China. She received her PhD degree in School of Mathematical and Information Sciences from Coventry University, U.K., in 2002. Her current research interests include Risk and Reliability Assessment, Vehicular Networks Security and Privacy Protection and Social Networks Security Management. Her research work has been supported by National Science Foundation of China.

Haojin Zhu (IEEE M'09-SM'16) received his B.Sc. degree (2002) from Wuhan University (China), his M.Sc. degree (2005) from Shanghai Jiao Tong University (China), both in computer science and the Ph.D. in Electrical and Computer Engineering from the University of Waterloo (Canada), in 2009. Since 2017, he has been a full professor with Computer Science department in Shanghai Jiao Tong University. His current research interests include network security and privacy enhancing technologies. He published more than 40 international journal papers, including JSAC, TDSC, TPDS, TMC, TWC, TVT, and 60 international conference papers, including ACM CCS, ACM MOBICOM, ACM MOBIHOC, IEEE INFOCOM, IEEE ICDCS. He received a number of awards including: IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award (2014), Top 100 Most Cited Chinese Papers Published in International Journals (2014), Supervisor of Shanghai Excellent Master Thesis Award (2014), Distinguished Member of the IEEE INFOCOM Technical Program Committee (2015), Outstanding Youth Post Expert Award for Shanghai Jiao Tong University (2014), SMC Young Research Award of Shanghai Jiao Tong University (2011). He was a co-recipient of best paper awards of IEEE ICC (2007) and Chinacom (2008), IEEE GLOBECOM Best Paper Nomination (2014), WASA Best Paper Runner-up Award (2017). He received Young Scholar Award of Changjiang Scholar Program by Ministry of Education of P.R. China in 2016.

Cailian Chen (IEEE S'03-M'06) received the B.Eng. and M.Eng. degrees in Automatic Control from Yanshan University, P. R. China in 2000 and 2002, respectively, and the Ph.D. degree in Control and Systems from City University of Hong Kong, Hong Kong SAR in 2006. She joined Department of Automation, Shanghai Jiao Tong University in 2008 as an Associate Professor. She is now a Full Professor. Before that, she was a senior research associate in City University of Hong Kong (2006) and postdoctoral research associate in University of Manchester, U. K. (2006-2008). She was a Visiting Professor in University of Waterloo, Canada (2013-2014). Prof. Chen has worked actively on various topics such as wireless sensor networks and industrial applications, computational intelligence and distributed situation awareness, cognitive radio networks and system design, Internet of Vehicles and applications in intelligent transportation, and distributed optimization. She has authored and/or coauthored 2 research monographs and over 100 referred international journal and conference papers. She is the inventor of more than 20 patents. Dr. Chen received the prestigious "IEEE Transactions on Fuzzy Systems Outstanding Paper Award" in 2008, and "Best Paper Award of The Ninth Int. Conference on Wireless Communications and Signal Processing" in 2017. She won the First Prize of Natural Science Award twice from The Ministry of Education of China in 2006 and 2016, respectively. She was honored "Changjiang Young Scholar" by Ministry of Education of China in 2015 and "Excellent Young Researcher" by NSF of China in 2016. Prof. Chen has been actively involved in various professional services. She serves as Associate Editor of IEEE TVT, PPNA (Springer), The World Scientific Journal: Computer Science, and ISRN Sensor Networks. She also served as Guest Editor of IEEE TVT, Symposium TPC Co-chair of IEEE Globecom 2016 and VTC2016-fall, Workshop Co-chair of WiOpt'18, and TPC member of many flagship conferences including IEEE Globecom, IEEE ICC, IEEE VTC, ICCVE and IEEE WCCI.

Kaoru Ota was born in Aizu-Wakamatsu, Japan. She received M.S. degree in Computer Science from Oklahoma State University, USA in 2008, B.S. and Ph.D. degrees in Computer Science and Engineering from The University of Aizu, Japan in 2006, 2012, respectively. She is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. From March 2010 to March 2011, she was a visiting scholar at University of Waterloo, Canada. Also, she was a Japan Society of the Promotion of Science (JSPS) research fellow with Kato-Nishiyama Lab at Graduate School of Information Sciences at Tohoku University, Japan from April 2012 to April 2013. Her research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems. Dr. Ota has received best paper awards from ICA3PP 2014, GPC 2015, IEEE DASC 2015, IEEE VTC 2016-Fall, FCST 2017, 2017 IET Communications Premium Award and IEEE ComSoc CSIM Best Conference Paper Award 2018. She is an editor of IEEE Transactions on Vehicular Technology (TVT), IEEE Communications Letters and the recipient of IEEE TCSC Early Career Award 2017.



Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently an Associate Professor in the Department of Information and Electronic Engineering at the Muroran Institute of Technology, Japan. He was a JSPS Research Fellow with School of Computer Science and Engineering, The University of Aizu, Japan and was a visiting scholar with BBCR group at University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong was selected as a Foreigner Research Fellow (a total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. His research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems. He has received best paper awards from IEEE HPCC 2008, IEEE ICESS 2008, ICA3PP 2014, GPC 2015, IEEE DASC 2015, IEEE VTC 2016-Fall, FCST 2017, 2017 IET Communications Premium Award and IEEE ComSoc CSIM Best Conference Paper Award 2018. Dr. Dong serves as an Editor for IEEE Transactions on Green Communications and Networking (TGCN), IEEE Communications Surveys and Tutorials, IEEE Network, IEEE Wireless Communications Letters, IEEE Cloud Computing, IEEE Access. He has been serving as the Vice Chair of IEEE Communications Society Asia/Pacific Region Information Services Committee and Meetings and Conference Committee, Leading Symposium Chair of IEEE ICC 2019, Student Travel Grants Chair of IEEE GLOBECOM 2019. He is the recipient of IEEE TCSC Early Career Award 2016, IEEE SCSTC Outstanding Young Researcher Award 2017, The 12th IEEE ComSoc Asia-Pacific Young Researcher Award 2017 and Funai Research Award 2018. He is currently the Member of Board of Governors and Chair of Student Fellowship Committee of IEEE Vehicular Technology Society.