Location privacy-preserving in online taxi-hailing services

Xiaoying Shen · Licheng Wang · Qingqi Pei · Yuan Liu · Miaomiao Li

Received: 4 March 2020 / Accepted: 4 August 2020 / Published online: 25 August 2020
© The Author(s) 2020

Abstract
Online taxi-hailing has become people’s most popular trip mode due to its convenience and low cost. However, it also poses a privacy threat to passengers and drivers, since the online taxi-hailing service providers are able to track their precise mobility trajectories. In addition, there is a certain time delay between the time of a passenger makes a request and the time of the driver arrives the passenger’s boarding position in current online taxi-hailing system. To solve these two problems, we present a new and efficient location privacy protection scheme based on the MinHash algorithm (LPPM). With the LPPM, the exact positions of passengers and drivers are generalized into a set of points of interest around them, and the distance between them is transformed into the similarity between the two sets. Thus a service provider can efficiently match passengers and drivers by using MinHash algorithm without revealing their specific location information. In this paper, we use mobile edge computing technology in the online taxi-hailing system to address the second challenge. It can speed up data processing, drivers can make decisions in advance and reduce the possibility of road congestion. Security analysis shows that LPPM has high security, and the final experimental results confirmed that LPPM is effective.

Keywords Location privacy protection · MinHash algorithm · Jaccard similarity · Online taxi-hailing · Mobile edge computing

1 Introduction
As an innovative industry in the context of “Internet +”, the online taxi-hailing service is constantly changing the way we travel. It offers personalized immediate auto services that fulfill the ever-increasing demands of the people. Not only does the emergence of online taxi-hailing service make people’s daily travel more controllable and convenient, but also it cuts the travel cost as well as improves travel efficiency, and to some extent, relieves the daily traffic pressure of the city.

The online taxi-hailing service brings great convenience to people’s travel, however, it also brings serious privacy threats, which provides an opportunity for malicious attackers to commit illegal acts. Location information is extremely sensitive to everyone, because it usually implies a large number of personal information, including religious beliefs, health problems, social relations, occupations, habits and so on. In the meanwhile, the online taxi-hailing service provider can track precise trajectories of all passengers and drivers on a large scale, and accumulate their historical trajectories. In case these data are acquired by malicious platforms, they may infer more privacy information from location tracking. For instance, a person frequently visits the hospitals will release his/her health condition. The disclosure of such sensitive information will pose a huge threat to user’s privacy. Accordingly, how to use online taxi-hailing services without exposing its specific location has attracted more and more attention.

In this paper, we propose a method to replace GPS information with a set of point-of-interest (POIs) [1] surrounding the user’s exact location for the subsequent calculations. To some degree, processing the passenger’s position information in this way protects the location privacy of passenger. Besides, in order to completely hide the exact position, the MinHash algorithm is introduced, which can transform the distance between passengers and drivers into the coincidence degree of two sets, the higher the coincidence is, the closer the distance
between them becomes. The function of encryption and dimensionality reduction of MinHash can significantly improve the computing efficiency and effectively protect the user’s location privacy. In addition, in the taxi-hailing system, users have a very high demand for real-time location information, if these data are transmitted to the cloud computing center, it will cause a waste of bandwidths and delay. In this paper, we use mobile edge computing technology [2] in the online taxi-hailing system to address this challenge. It can speed up data processing, drivers can make decisions in advance and reduce the possibility of road congestion.

In summary, our main contributions are listed as follows:

- A new and efficient ride-matching protocol is proposed to realize online taxi-hailing services. To preserve passengers’s and drivers’s location privacy, we use a set of POIs to blur their exact location. Then, converting the distance between the passenger and the driver into the Jaccard similarity of the two sets.
- To meet the real-time requirement of traffic information in the taxi-hailing system, we outsource some complicated calculation to edge devices, reducing the time it takes to manipulate the location data. Consequently, the drivers can make timely decisions and the possibility of road congestion can be reduced.
- We evaluate the performance of proposed algorithm theoretically and experimentally. Theoretical analysis and experimental results showed that our scheme achieves both high matching accuracy and efficiency.
- Compared with the work of pham et al. [3], the scheme presented in this paper has significant reduction in both computational cost and communication cost.

The rest of this paper is organized as follows: Section 2 discusses the related works on location privacy-preserving in vehicle networking. Section 3 briefly recalls some concepts and tools that need to be used in this article. Section 4 proposes the adversarial model and the model of a complete location privacy preservation based on the MinHash algorithm. In Section 5, we describe the ride-matching protocol in more detail. We give the security analysis of LPPM in Section 6. Section 7 describes the experimental methodology to evaluate the performance of LPPM scheme. Finally, we give concluding remarks in Section 8.

2 Related works

In this section, we survey related work, focusing on location privacy protection algorithms in vehicle networking, prior works include Mix-zone, K-anonymity, homomorphic cryptography and differential privacy.

2.1 Mix-zone

As early as 2003, Beresford et al. [4] proposed the pseudonym mechanism to protect the privacy of users. The idea was to divide the space into mixed area and application area. Users of the application area can send location-based services requests and receive services, users of the mixed zone are prohibited from using location-based services. At the same time, users must change to a new pseudonym before leaving the mixed zone to protect the location privacy. In 2005, by changing the original idea that pseudonym must be replaced within a specific geographical areas (Mix-zone), Raya et al. [5] redefined the Mix-zone pseudonym mechanism which determines the replacement frequency of pseudonyms with speed. In 2007, Freudiger et al. [6] proposed a cryptographic mix-zones (CMIX) scheme. The roadside equipment is required to distribute symmetric keys to vehicles and encrypt messages to protect location privacy. Since then, Palanisamy et al. [7] have improved anonymity on the basis of mix-zone scheme to better protect the location privacy of car owners in the process of communication. At the same year, Ying et al. [8] proposed a location privacy protection scheme for taxi-hailing system based on candidate location list, which dynamically changed Mix-zone to improve anonymity. The disadvantage of mixed zone technology is to limit users to use location-based services in the mixed zone, some services that need to provide location information continuously cannot be used, such as navigation services.

2.2 K-anonymity

The concept of K-anonymous model was proposed by Samarati and Sweeney [9] for the first time. Gruteser et al. [10] firstly used it in the location privacy protection scheme and proposed the location K-anonymity. Its core idea is to choose a block to replace the exact location of a certain vehicle, meanwhile it requires at least K indistinguishable vehicles to drive in this block. In this way, if a car in this block initiates a request for location-based services, the location server cannot identify the real service requestor, because of there is another K−1 cars driving in this block, thereby the protection of the location privacy is realized. Yang et al. [11] made some improvements on the basis of Gruteser’s scheme. In 2011, Chow et al. [12] used spatial cloaking thoughts, the P2P network is used to find the cloaked area which is satisfied K-anonymous, and a node is randomly selected in the anonymous area as the author of the query to send the service request, achieving the effect of hiding the actual query node with a fuzzy area. In 2018, Cui et al. [13] proposed a location privacy protection scheme that uses virtual location and path confusion to protect real-time location data in vehicle ad hoc networks based on the
basic idea of K-anonymity. The vehicle can dynamically generate a virtual location based on the condition of the surrounding vehicle, providing misleading information about the driving route, thereby achieving location privacy. However, this method lacks diversity in sensitive data or faces the risk of privacy exposure when the attacker has auxiliary information.

### 2.3 Homomorphic cryptography

The third technique is homomorphic cryptography. In 2014, Xi et al. [14] applied Private Information Retrieval (PIR) technology and additive homomorphic encryption technology to road routing to calculate the shortest path, meanwhile protecting location privacy. In 2017, Pham et al. [3] proposed a privacy protection scheme based on somewhat homomorphic encryption. Service providers can match passengers and drivers without knowing the user’s identity and location information. However, homomorphic encryption needs a lot of computing time and communication cost, which can not meet the user’s fast corresponding needs.

### 2.4 Differential privacy

The forth technique is differential privacy. Dwork [15] proposed the differential privacy model in 2008. Miguel et al. [16] firstly applied this model to the field of location privacy protection. The core idea of this mechanism is to achieve geo-indistinguishability by adding some controllable random noise to the user’s precise location.

The motivation of our work is based on generalization and cloaking techniques, which are the key technologies to achieve K-anonymity. Inspired by them, the positions of the passenger and the driver are expressed as two independent sets, and the distance between the passenger and the driver is converted into the Jaccard similarity of the two sets. Naturally, we can use MinHash algorithm proposed in [17] to quickly estimate the similarity of two sets. According to the distance-keeping property of Minhash, our transformation is reasonable. In the online taxi-hailing system, by using this approach the server can match passengers and drivers efficiently and accurately while keeping their locations privacy against the server and outsider.

### 3 Preliminaries

In this section, we briefly introduce some tools Mobile edge computing, Jaccard similarity, MinHash algorithm, Non-interaction key exchange which are employed in our solutions.

#### 3.1 Mobile edge computing

Mobile edge computing [2] technology is an important technology to improve the application capability of service platforms in the future 5G technology. The basic idea is to move the cloud computing platform from the inside of the mobile core network to the edge of the mobile access network to enable mobile subscribers to access IT and cloud computing services at the close proximity. This concept deeply integrates the traditional telecom cellular network with the Internet business, aiming to reduce the end-to-end delay of mobile service delivery and explore the inherent capabilities of wireless network, thus creating a telecom level service environment with high performance, low latency and high bandwidth, accelerate the rapid download of various contents, services and applications in the network. Further more, consumers can enjoy uninterrupted high-quality network experience.

#### 3.2 Jaccard similarity

The Jaccard similarity[18] is a common indicator of the similarity between two sets. Let $X$ be a set, $A$ and $B$ be subsets of $X$, then the Jaccard similarity coefficient is defined as follows:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where $|A \cap B|$ denotes the number of elements of their intersection and $|A \cup B|$ denotes the number of elements of their union.

#### 3.3 MinHash algorithm

In 1997, Broder et al. [17] proposed the concept of MinHash to quickly estimate the similarity between two text documents. It was originally used in the AltaVista search engine to detect and eliminate duplicate Web pages in search results. Nowadays, it is widely used in similarity retrieval, recommendation system and cluster analysis of large data sets.

The core idea of the MinHash algorithm is to replace the problem of calculating the similarity of the two sets $A$ and $B$ with the similarity of the two MinHash signatures $h_{min}(A)$ and $h_{min}(B)$. The computation process of MinHash-based Jaccard similarity usually involves three main phases: the shingling phase represents the textual documents into set; the MinHashing phase generates the relevant MinHash signature; the last phase is approximate computation. The specific process is shown in Fig. 1. First of all, the first document $1$ is divided into $n$ parts and represented as the set $A = \{a_1, a_2, \cdots, a_n\}$. Secondly, $s$ random permutations
are selected and orderly acted on the set $A$ to generate $h_i(A) = \{h_i(a_1), h_i(a_2), \cdots, h_i(a_n)\}, i \in [1, \cdots, s]$ and then the minimum hash value of $h_i(A)$ is the $i$-th element of the MinHash signature $h_{\text{min}}(A)$. Take the same action on the second file 2 to get the MinHash signature $h_{\text{min}}(B)$.

Now, the similarity between file 1 and file 2 can be estimated by the similarity of their respective minimum hash signatures, i.e.

$$J(A, B) = \frac{|h_{\text{min}}(A) \cap h_{\text{min}}(B)|}{s}.$$ 

### 3.4 Non-interactive key exchange

In this paper, we choose traffic administration bureau as the private key generator (PKG), it is a trusted authority. When a user wants to sponsor a taxi-hailing service, he needs to request his private key from PKG. The non-interaction key exchange protocol [19] can be naturally divided into four distinct algorithms: Setup, Master-key generation, Private key distribution and Shared key computation.

- **Setup**: The PKG chooses a prime $q$ and two groups $G_1$ and $G_2$ of order $q$, where $G_1$ is an additive group and $G_2$ is a multiplication group. Let $e : G_1 \times G_1 \rightarrow G_2$ be an efficiently computable bilinear pairing and $H : \{0, 1\}^* \rightarrow G^*$ be cryptographic hash functions. All these parameters are publicly known.

- **Master-key generation**: The PKG chooses a random master-key $x \in \{1, \cdots, q - 1\}$.

- **Private key distribution**: whenever a user $U$ firstly wishes to use the system, he contacts the PKG and asks for his private key. Using $U$’s identity $id_U$, the PKG computes $U$’s private key $sk_U = xH(id_U) \in G$ and sent it to $U$.

- **Shared key computation**: Suppose that users $U_1$ and $U_2$ wish to create a common secret key. $U_1$ computes $U_2$’s public key $pk_{U_2} = H(id_{U_2}) \in G$ and shared key $k_{U_1} = e(xH(id_{U_1}), H(id_{U_2}))$. And conversely $U_2$ computes $U_1$’s public key $pk_{U_1} = H(id_{U_1}) \in G$ and shared key $k_{U_2} = e(xH(id_{U_2}), H(id_{U_1}))$.

The bilinearity of $e$ makes it easy to see that $k_{U_1} = k_{U_2}$ in fact, and thus constitutes a secret shared key between $U_1$ and $U_2$.

### 4 Problem statement

#### 4.1 System model

There exists such a common phenomenon in the current online taxi-hailing system that the service providers can track the precise trajectories of all passengers and drivers. It makes the online taxi-hailing service face a serious threat to privacy and security. If the disclosure of location privacy can’t be solved as quickly as possible, more and more users will not continue to use such services. The online taxi-hailing dispatch system designed in this paper mainly consists of drivers, passengers and service provider. The passengers are usually smartphone users who need to take a taxi, and their main operation is to send ride requests. The drivers are ordinary taxi renters, their main operation is to send order requests. The service provider receives a ride request from a passenger and matches the request with the driver who can provide the service. His main operation is ride matching. In this paper, we combine the current environment of online taxi-hailing, and propose a location privacy protection scheme based on MinHash algorithm without revealing the location information of passengers and drivers. It strengthens the confidentiality of location data in the taxi-hailing service and improves the location privacy protection ability of the taxi-hailing service.

The core idea of LPPM is shown in Fig. 2. Passenger ($P$ for short) chooses POIs in a certain range around him (e.g. 1km) as the reporting location rather than his exact location, where 1km is the farthest distance between $P$ and $D$, this parameter can be determined by himself. Similarly,
the driver (D) also chooses POIs within a certain range (e.g. 3km) around him as his reporting location, where 3km indicates that D is willing to pick up passengers within 3km. In Fig. 2, the small circle represents the area with a radius of 1 km around P, the big circle represents the area with a radius of 3 km around D, and the small red dot represents POI. In Fig. 2, we only draw a few POIs, but in fact the number of POIs around a user can reach hundreds or thousands. All subsequent operations are based on the POIs of P and D. Therefore, when passengers want to reserve a taxi through the online taxi-hailing system, the system can’t directly uses the passenger’s accurate location information to perform vehicle scheduling. To a certain extent, it protects the location information of passengers and drivers, and achieves the rapid matching between passengers and drivers.

The design of LPPM is based on the assumption: the set constituted by all POIs in the small circle of Fig. 2 is denoted as the features of passengers. The set of POIs in the big circle is denoted as the driver’s features. The more identical feature points of small circle and big circle have, the closer P and D are. That is to say, the matching degree between small circle and big circle is higher, vice versa. Therefore, in the next discussion of this paper, the distance between P and D will be measured by the matching degree between their corresponding feature sets. LPPM is designed as follows.

- First of all, the passenger P sets a radius r, and chooses a region with the radius r and centered on himself, then some POIs in this area are selected as his feature set. Similarly, D sets the radius and selects POIs to get D’s feature set. SP publishes an ordered set of hash functions which are used to encrypt and reduce the dimension of feature sets. These functions should be updated periodically.
- Step 1: When D prepares to carry passengers, he/she enters the listening state, and calculates his/her signature vector in real time according to the set of ordered hash functions released by SP and sends his/her signature vector to SP;
- Step 2: When P wants to take a taxi, he/she needs to calculate his signature vector according to the set of ordered hash functions published by SP, and then sends it to SP;
- Step 3: SP stores the received signature vector of D in the local database and updates it in real time. When SP receives P’s signature vector, it uses the formula of Jaccard similarity to calculate the similarity between P and D.
- Step 4: SP selects a pair of passengers and drivers with the highest similarity, and sends the phone number to both parties.
- Step 5: P and D establish a secure channel by using ID-based non-interactive key agreement protocol to exchange specific locations with each other. If necessary, this step can be outsourced to mobile edge devices.

Figure 3 is an image description of the above scheme. Based on a set of ordered hash functions published by SP, passengers and drivers can calculate signature vectors from feature sets, this computing process can be regarded as feature extraction. Note that the number of signature sets is much smaller than the number of feature sets, the hash function used here not only plays the role of encryption, but also plays the role of dimension reduction, it greatly improves the computational efficiency.
4.2 Threat model

Our goal is to propose an online taxi-hailing system that can protect both passenger’s and driver’s privacy. To do so, we suppose that our system consists of passengers, drivers and service provider (SP), where drivers and passengers are active adversaries, SP is the passive adversary. This assumption is reasonable, because SP will not actively disclose user’s privacy information for its own reputation and interests. In addition, we assume that almost all drivers and passengers will not collaborate with service providers, because in taxi-hailing service, passengers and drivers are not SP’s employees.

Literature [20] gives a threat taxonomy for privacy threats in the current online taxi-hailing system. By following the OWASP risk-rating methodology to show the risk levels of each privacy threat in Table 1.

Table 1 The types of attacks

| Description | Privacy threat                        | Risk  |
|-------------|--------------------------------------|-------|
| O → P       | PII harvesting                       | medium|
| SP → P      | location tracking                    | High  |
| D → P       | PII harvesting                       | Low   |
| O → D       | location tracking                    | High  |
| SP → D      | PII harvesting                       | medium|
| P → D       | PII harvesting                       | Low   |
| O → SP      | PII harvesting and ride data breach  | High  |

X → Y: X attacks Y

Table: PII: Personally identifiable information

Although the threats in Table 1 are not exclusive to online taxi-hailing service, however, due to the limitations of the design of online taxi-hailing system, the platform can collect more sensitive information. As a result, the risk of some threats has increased significantly. In this work, we focus on the high risk threats, and to reduce these risk to a level that is at least as low as that of taxi services.

**SP → P location tracking**: Compared with traditional taxis, online taxi-hailing service providers can track the precise trajectory of passengers in real time during their rides and accumulate their historical activity trajectories. User’s privacy information can be easily inferred from such data by malicious platforms or users who can access these data. For example, a person’s health can be inferred from the fact that he stayed in the clinic for several days; two people often appear in the same place at the same time, it can be inferred that they are close to each other.

**O → D PII harvesting**: At present, in order to make passengers enjoy better service, the most online taxi-hailing platforms will reveal the driver’s name, phone number, current location, and car plate number, and even some service providers also reveal the driver’s photo and car model and picture. A malicious outsider could collect these information by using multiple fake accounts, selecting different pickup locations, sending request and then canceling.

**O → SP PII harvesting and ride data breach**: An active adversary will do his best to collect personal information of passengers and drivers for some business purpose.
Because online taxi-hailing service providers can collect personal information of passengers on a large scale, including identity information, location information and mobile trajectory. Once $SP$ is attacked and the information is used by criminals, it will greatly damage the personal privacy and property security of users.

5 Design goals

Under the system model and threat model, a location privacy-preserving scheme in online taxi-hailing services should satisfy the following design goals.

- Accuracy: The location-based services provider should be able to find the nearest driver for a passenger with a high probability.
- Efficiency: The proposed scheme should be efficient for practical application.
- Security: The location-based services providers and external attackers should learn nothing about the locations of the passengers and drivers.

6 Ride-matching protocol

In this section, we describe in detail the location privacy protection scheme in our online taxi-hailing service.

6.1 The calculation of the MinHash algorithm

The MinHash algorithm is mainly used for the similarity comparison of two sets. Its principle is that the probability that the MinHash values of a random row arrangement of two sets are equal is equal to the Jaccard similarity of the two sets. Usually, by traversing all the elements in the two sets and counting the number of the same elements in the two sets, the similarity of the two sets is expressed. When the number of elements in these two sets is unusually large, and there are many sets to judge the similarity between two sets, the traditional method can become very time-consuming. For this purpose, we need to define a kind of hash function that maps a 32-bit integer to another 32-bit integer, and it can simulate the effect of a random permutation.

Randomly selecting $s$ hash functions $h_1, h_2, ..., h_s$ with the form of $h(x) = (ax + b) \mod c$, where the variable $x$ is the input of the hash function. The coefficients $a, b \in \mathbb{Z}$ are randomly chosen and less than the maximum value of $x$. The $c$ is a prime that is slightly larger than the maximum value of $x$, and such value can ensure that the hash function has a small collision probability, and the impact of small collision probability on a very large data set can be ignored.

Therefore, different hash functions can be obtained by selecting different $a$ and $b$, that is to say, different mappings can be obtained by selecting different $a$ and $b$. Now, we act $h_1$ on the original feature set to get different hash values, and then select the smallest hash value as the first element of the signature vector. Performing the same operation on original feature set. By making use of $h_2, ..., h_s$, we can get the second element, $\cdots$, the $s$-th element of signature vector, and placing all the elements in an ordered list, we get the signature vector with $s$ elements. Usually $s$ is much smaller than the length of the original feature set.

Note that in LPPM, $P$ and $D$ firstly select POIs around them to get the original feature set, and then perform the MinHash algorithm to extract signature vector from the original feature set and send it to the taxi-hailing service provider. When it receives the signature vectors of passengers and drivers, it calculates the Jaccard similarity between the two vectors. 1-Jaccard similarity is the Jaccard distance between $P$ and $D$. Obviously, the larger the value of $s$ is, the more elements the signature vector has, and the similarity between signature vectors will be closer to the actual Jaccard similarity of the original feature set. In addition, the number of elements in signature vector is much smaller than that of original feature set, so it is much simpler to calculate the similarity between signature vectors, and the computing efficiency of the platform is significantly improved.

6.2 User registration

We assume the transportation bureau issued a pair of public and private keys for each passenger and driver during the registration phase. Each new user sends his or her identity information (e-mail or ID card number or telephone number) to the PKG to apply for his or her public/private key ($pk = H(id)$, $sk = x \cdot H(id)$) before using the service, a private key $sk$ that is kept private locally, and a public key $pk$ that can be publicly released. Anyone can obtain a copy of the public key on the Internet. Here, the role of the public key and the private key is to negotiate the shared key, thereby to establish a secret channel between the driver and the passenger, through this channel the real-time location of the driver can be transmitted to the passenger.

6.3 Log in to the service

To use the service, passengers and drivers need to register a taxi-hailing APP with their phone numbers. The platform verifies the uniqueness of the phone number by sending a verification code. If authentication is successful, the phone number is stored on $SP$’s database encrypted and used as the unique identifier for the user.
6.4 Map segmentation

The location-based services provider divides the map into $M$ location units and requests the user to be located in one of them. As shown in Fig. 4, dividing the map into $8 \times 8$ areas, that is, $M = 64$, each zone representing a location unit, that is to say, $Map = \{C_1, C_2, \ldots, C_{64}\}$. In this paper, we agree that before execution of any private ride-matching protocol, the system will automatically divide the map of the region.

6.5 Design of LPPM scheme

Once a passenger logs into the client on his mobile phone, he can initiate a ride request, the operations are performed by the passenger, the driver and the SP as shown in the following and Table 2 shows some notations of Fig. 5.

- step 1: The passenger $P$ selects some location unit near him to form his own feature set $P(p_1, p_2, \ldots, p_n)$. Similarly, $D$ gets his feature set $D(d_1, d_2, \ldots, d_n)$. $SP$ publishes a set of hash functions $\{h_i\} (i = 1, 2, \ldots, K)$ and updates them periodically, which is used to encrypt and reduce the dimension of feature sets to obtain signature vectors.

- Step 2: When $D$ prepares to carry passengers, it enters the “listening” state, and calculates its MinHash signature $Dsig(d_1, d_2, \ldots, d_s) (s << n)$ in real time according to the hash functions released by $SP$, then sends $Dsig(d_1, d_2, \ldots, d_s)$ to $SP$.

- Step 3: When $P$ wants to take a taxi, he needs to calculate his signature vector $Psig(p_1, p_2, \ldots, p_n) (s << n)$ according to the hash functions published by $SP$, and then sends $Psig(p_1, p_2, \ldots, p_n)$ to $SP$.

- Step 4: $SP$ stores the received signature vector $Dsig(d_1, d_2, \ldots, d_s)$ of $D$ in the local database and updates it in real time. When $SP$ receives $P$’s signature vector $Psig(p_1, p_2, \ldots, p_n)$, it uploads the latest driver’s signature vector and passenger’s signature vector to the cloud server to calculate a pair of passenger and driver with highest Jaccard similarity. And then $SP$ sends a virtual phone number $TD$ associated with the driver’s phone number to the passenger. At the same time, $SP$ also sends a virtual phone number $TP$ associated with the passenger’s phone number to the driver.

- step 5: The passenger $P$ randomly generates two random numbers $r_1$ and $r_2$, and sends the pair $(r_1 \cdot xH(TP), r_2H(TD))$ to the roadside unit. The roadside unit calculates $K'_P = e(r_1 \cdot xH(TP), r_2H(TD))$ with its powerful local computing power and returns the result to passenger. And then $P$ gets the shared secret key $K_P = KDF((K'_P)^{r_1})$ by using the key derivation function (KDF) [21]. The driver use the on-board sensors to calculate $K_D = KDF(e(x \cdot H(TD), H(TP)))$. Obviously, $K_0 = K_P = K_D$ is the shared key of passenger and driver.

| Table 2 Table of notation for Fig. 5 |
|--------------------------------------|
| Notation | Description |
| $h_i (i = 1, 2, \ldots, s)$ | hash function |
| $Psig$ | Signature vector of passenger |
| $Dsig$ | Signature vector of driver |
| $TP$ | Phone number of passenger |
| $TD$ | Phone number of driver |
| $H$ | A cryptographic hash function |

Fig. 4 Map segmentation

Fig. 5 LPPM Scheme. The solid arrows represent common channel and dashed arrows represent secure channel
step 6: The passenger and the driver establish a secure channel via the shared key $K_0$. With this secure channel, the passenger selects randomly $K_1 \in \mathbb{Z}_q$ and computes $K = E_{K_0}(K_1)$, then sends $K$ to the driver. When the driver receives the passenger’s encryption key $K$, he decrypts $K$ to get the secret key $K_1$. $K_1$ is their shared key in subsequent communications.

step 7: The passenger encrypts the current precise location $L$ with the shared key $K_1$ in real time to generate the ciphertext $C = E_{K_1}(L)$, and passes $C$ to the driver through the secure channel. When the driver receives ciphertext $C$, the driver recovers the precise location $L$ of the passenger by calculating $L = D_{K_1}(C)$. And then drive to this location to pick up the passenger.

step 8: When the driver arrives the pick-up location, he sends his license plate number to the passenger so that the passenger can identify the correct driver.

7 Security analysis of LPPM

In this section, we analyze the security of the LPPM scheme in detail, and show that it can effectively resist the attack proposed in Part 4.2. We mainly analyze three types of attacks with higher risk in this paper.

SP → P/D location tracking: In the protocol request phase, passengers and drivers calculate the signature vector according to their own location feature set and send them to the platform, the platform stores signature vector for calculating the Jaccard distance between passengers and drivers. The signature vector is obtained by applying a set of random hash functions to some POIs near the specific location of the passenger/driver. According to the unidirectivity of the hash function, even if the platform gets the signature vector, it can not infer the specific location of passenger/driver. Once the right driver is matched for the passenger, the passenger will not report his position to the platform. The platform will not be able to track the trajectory of passengers.

O → D harvesting: Under the current taxi-hailing system, a malicious outsider could disguise as legitimate passenger to harvest the driver’s personal information, by making requests and then canceling the ride requests. With LPPM, before arriving at the passenger’s pick-up position, the passenger only knows the driver’s virtual phone number. The malicious passenger could not infer the driver’s personal information from the phone number. When the driver arrives near the pick-up location reported by passenger, the passenger only knows the driver’s license plate number. It’s the same as taking a regular taxi.

Outsider → SP PII harvesting and ride data breach: From the design details of the LPPM scheme, we can clearly see that the platform only stores the signature vectors and encrypted phone numbers of passengers and drivers. If there is no private key for the platform, no information about passengers or drivers could be obtained by an encrypted phone number. This is reasonable because the platform will not disclose its private key for its own business interests, thus the sensitive information only be leaked through the signature vector. But for the attacker, it is difficult to refer personal information for $P$ and $D$ when the signature vectors of $P$ and $D$ are known. Therefore, the security of LPPM is high. In addition, the purpose of calculating the similarity between the signature vectors is to obtain two sets with the highest similarity, namely, to find the $P$ and $D$ with the closest Jaccard distance. However, the Jaccard distance is only a fraction within the range of $[0, 1]$, it has no practical significance and can’t represent the actual distance between $P$ and $D$. Therefore, an attacker can’t estimate the specific location of $P$ and $D$ by selecting multiple similar values of different points with the same user.

8 Performance analysis

At first, a theoretical analysis on the performance of LPPM scheme, from the perspectives of computing cost, storage cost and communication cost, is given in Table 3.

- Communication overhead: In the proposed scheme the passenger and the driver calculate their signature vectors respectively and send them to the location-based services provider. The location-based services provider calculates the Jaccard similarity between empty car drivers in the current zone and the passenger, and then selects the pair of passenger and driver with the highest Jaccard coefficient, and sends the phone numbers of each other to both parties, the above process involves a communication between three entities.

- Computation overhead: In the above communication, we need 1 encryption, 1 decryption, 2 multiplication, $(s \log s + sn \log n)$ addition, 1 bilinear pair, 2 cryptographic hashing and $sn$ MinHashing. Where $s$ is the number of MinHash functions, and $n$ is the number of empty drivers in the current zone.

This paper uses the same method as Pham [3] to establish the secure channel, so this paper mainly compares the communication cost and calculation cost of the stage from the passenger’s request to the matching success. The performance comparison between our scheme and Pham scheme is shown in Table 4.

Next, we conduct simulations on LPPM by the following steps:
Table 3  Efficiency analysis of LPPM

| Roles          | Operations                        | Complexities                        |
|----------------|-----------------------------------|-------------------------------------|
|                |                                   | Computation cost | Communication cost |
| PKG            | Extract private key               | $1H_c + 1 \otimes_G1$              | 256bit             |
| Platform       | Issue hash function               | $[s \log s] \oplus_Zq + 1 \otimes_Zq$ | 40bit              |
| Driver/Passenger| Compute $J_{sim}$                  | $1H_c + 1 \otimes_G1$              | 256bit             |
|                | Register                          | $[sn]H_{m} + [sn \log n] \oplus_Zq$ | 4sbit              |
|                | Compute signatures                | $1H_c + \hat{e}$                   | 256bit             |
| Encryption/Decryption |                          | $E_{SM4}/D_{SM4}$                  | 256bit             |

$H_c$: Cryptographic hashing  
$H_m$: Min-hashing  
$\hat{e}$: Bilinear pairing  
$\oplus_X$: addition/substraction/comparison over $X$ (where $X$ could be instantiated as $Z_q$, $G_1$ and $G_2$, respectively)  
$\otimes_X$: multiplication/division/inversion over $X$  
$E_{SM4}/D_{SM4}$: Encryption/decryption of SM4

8.1 Scenario settings

In this simulation, we select a relatively simple scenario. One passenger $P$ and 10 drivers $D_i(i = 1, 2, \cdots, 10)$ are randomly deployed in the target area. The service provider randomly publishes a set of hash functions $h_i(i = 1, 2, \cdots, s)$ and updates them every minute. Each driver sets the maximum distance he can pick up passengers to 3 km. Within this range, he selects $N$ feature points as his feature set and calculates his signature vector, and then reports it to the service providers. When a passenger wants to take a taxi, he logs in to his mobile App on his smartphone and sets the range of feature points that can be selected to be 1 km. In this range, $N$ feature points are selected as his feature set. He calculates his signature vector and reports it to the service provider. It is up to the service provider to calculate the match degree between the two signature vectors.

8.2 Dataset

The experimental data set used in this paper is from the University of California Irvine Machine Learning Library (https://archive.ics.uci.edu/ml/index.php). The trajectory data of about 320 taxis of Porto, in Portugal, are collected as the experimental data set. Each data sample corresponds to one completed trip. It contains a list of GPS coordinates mapped as a string. Each pair of coordinates is identified by the same brackets as [Longitude, Latitude]. This list contains one pair of coordinates for each 15 seconds of trip. The last list item corresponds to the trip’s destination while the first one represents its start.

8.3 Experimental equipment

The experimental equipment is a Lenovo computer with a processor configured with an Intel(R) Core(TM) i7-6700M CPU @ 3.40 GHz.

8.4 Accuracy

The purpose of LPPM is to find the nearest driver for the passenger who requests a taxi. Because calculating the similarity between two sets by using the MinHash function yields an approximate result, matching errors may exist.

Table 4  Performance comparison

| Scheme                  | Number of communication | Computation time                                                                 |
|-------------------------|-------------------------|---------------------------------------------------------------------------------|
| Pham’s scheme [3]       | 4                       | $[4n + 1] \oplus_{R_q} + [2n + 1] \otimes_{R_q} + 4NTT^{-1} + 1NTT + 2E_H + 2D_H$ |
| Our scheme              | 1                       | $[s \log s] \oplus_Zq + 1 \otimes_Zq + [sn]H_{m} + [sn \log n] \oplus_Zq$        |

$E_H/D_H$: Encryption/decryption of somewhat homomorphic encryption  
$R_q$: A polynomial quotient ring  
NTT: Number-theoretic transform
However, MinHash may be seen as an instance of locality sensitive hashing, it has the good property of keeping the distance constant. Specifically, two points that are very close in a certain space $A$ are still close after being mapped to another space $B$ by the MinHash function, and vice versa. Therefore, it is reasonable to replace the Euclidean distance with the Jaccard distance. In the LPPM scheme, we firstly use the MinHash function to convert a large collection of driver’s or passenger’s location into a MinHash value, then to find the driver closest to the passenger by comparing the similarities between the two signature vectors. Of course, the accurate Jaccard similarity between the original sets can not be obtained by signature vectors, but the estimated results are consistent with the real ones. The greater the signature set is, the higher the accuracy of estimation is.

### 8.5 Efficiency

We evaluate the effect of the size of datasets on the performance of our scheme, we do so for five different sizes of the dataset: 100, 1000, 2500, 5000, 10000, this provides different levels of matching speed. Detailed results are shown in Fig. 6.

In Fig. 6, the blue line represents the time is taken to calculate the similarity of feature vectors, and the orange line represents the time is spent in calculating the similarity of signature vectors. It can be seen from Fig. 6 that when the scale of feature vectors increases, the former takes much more time than the latter, which means that the LPPM scheme is suitable for large-scale data computing scenarios. And the experimental results show that LPPM can improve the computational efficiency of Jaccard similarity of two sets, and further can significantly improve the matching speed of passenger and drivers.

![Fig. 6](image_url) The time of calculating the Jaccard similarity

### 9 Conclusion

In this paper, we first analysed the privacy threats in the current online taxi-hailing service system. In order to resist these threats, we proposed a location privacy protection scheme based on MinHash algorithm which can match passengers and drivers efficiently without leaking the privacy of passengers and drivers. The main idea of LPPM is to generalize the user’s exact location with a set of POIs around the user, thus converting the distance between the passenger and the driver into the Jaccard similarity between the two sets. And then use MinHash algorithm to compute the Jaccard similarity. This method can effectively hide the precise location of the user, but did not affect the calculation of distance between passengers and drivers. Furthermore, in order to meet the real-time requirement of traffic information in the taxi-hailing system, we applied the mobile edge computing technology to the Online taxi-hailing system to optimize the location-based service. Experimental results showed that our scheme achieves both high matching accuracy and efficiency. Compared to previous cryptographic scheme LPPM has acceptable performances in computational cost, storage cost and communication cost. Unfortunately, the scheme in this paper is based on the basic non-interactive key exchange protocol and cannot resist the man-in-the-middle attack. In the future, we will design a more secure location privacy protection scheme that can resist man-in-middle attacks. In addition, we plan to implement a complete system prototype on mobile platform, including fair payment and reputation management and other functions.

**Acknowledgments** This work is supported by the National Key Research and Development Program of China (2018YFE0126000), the Key Program of NSFC-Tongyong Union Foundation under Grant U1636209 and BUPT excellent Ph.D. students foundation (Grant No. CX2019233, CX2019119).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit [http://creativecommons.org/licenses/by/4.0/](http://creativecommons.org/licenses/by/4.0/).
References

1. Liu B, Fu Y, Yao Z, Xiong H (2013) Learning geographical preferences for point-of-interest recommendation. In: The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013. ACM, Chicago, pp 1043–1051
2. Beck MT, Werner M, Feld S, Schimper T (2014) Mobile edge computing: a taxonomy. In: Accepted for the sixth international conference on advances in future internet
3. Pham A, Dacosta I, Endignon G, Troncoso-Pastoriza JR, Huguennin K, Hubaux J (2017) Oriide: A privacy-preserving yet accountable ride-hailing service. In: 26th USENIX Security Symposium. USENIX Security 2017. Vancouver, pp 1235–1252
4. Beresford AR, Stajano F (2003) Location privacy in pervasive computing. In: IEEE Pervasive Computing, vol 2, pp 46–55
5. Raya M, Hubaux J (2005) The security of vehicular ad hoc networks. In: Proceedings of the 3rd ACM Workshop on Security of ad hoc and Sensor Networks, SASN 2005. ACM, Alexandria, pp 11–21
6. Freudiger J, Raya M, Félegyházi M, Papadimitratos P, Hubaux J-P (2007) Mix-zones for location privacy in vehicular networks. In: Win-ITS 07
7. Palanisamy B, Liu L (2015) Attack-resilient mix-zones over road networks: Architecture and algorithms. IEEE Trans Mob Comput 14:495–508
8. Ying B, Makrakis D (2015) Pseudonym changes scheme based on candidate-location-list in vehicular networks. In: 2015 IEEE International Conference on Communications. ICC 2015, London, pp 7292–7297
9. Samarati P, Sweeney L (1998) Generalizing data to provide anonymity when disclosing information (abstract). In: Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. ACM, Washington, pp 188
10. Gruteser M, Grunwald D (2003) Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the First International Conference on Mobile Systems, Applications, and Services, MobiSys 2003. USENIX, San Francisco
11. Yang D, Fang X, Xue G (2013) Truthful incentive mechanisms for k-anonymity location privacy. In: Proceedings of the IEEE INFOCOM, Turin, pp 2994–3002
12. Chow C, Mokbel MF, Liu X (2011) Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. Geoinformatica 15:351–380
13. Cui J, Wen J, Han S, Zhong H (2018) Efficient privacy-preserving scheme for real-time location data in vehicular ad-hoc network. IEEE Internet Things J 5:3491–3498
14. Xi Y, Schowiebert L, Shi W (2014) Privacy preserving shortest path routing with an application to navigation. Perv Mob Comput 13:142–149
15. Dwork C (2008) Differential privacy: A survey of results. In: Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008, Xi’an, China, April 25-29, 2008. Proceedings, vol. 4978 of Lecture Notes in Computer Science. Springer pp 1–19
16. Andrés ME, Bordenabe NE, Chatzikokolakis K, Palamidessi C (2013) Geo-indistinguishability: differential privacy for location-based systems. In: 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS’13, Berlin, pp 901–914
17. Broder AZ, Charikar M, Frieze AM, Mitzenmacher M (1998) Min-wise independent permutations (extended abstract). In: Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, pp 327–336
18. Ertl O Supeminhash-A new minwise hashing algorithm for jaccard similarity estimation. CoRR arXiv:1706.05608
19. Sakai R, Oghishi K, Kasahara M (2000) Cryptosystems based on pairing. Symp Cryptogr Inf Secur Jpn 45:26–28
20. Pham A, Dacosta I, Jacot-guillarmod B, Huguennin K, Hajar T, Tramèr F, Gligor VD, Hubaux J (2017) Privateride: A privacy-enhanced ride-hailing service, PoPETs, 2, 38–56
21. (cryptography) K (2007) key derivation function, pp 273

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Xiaoying Shen received the M.S. degree from Northwest Normal University in 2017, and she is currently a Ph.D candidate in Beijing University of Posts and Telecommunications. Her current research fields include modern cryptography, network security and blockchain technology.

Licheng Wang received the Ph.D. degree from Shanghai Jiao Tong University in 2007. His current research interests include modern cryptography, network security, trust management, etc. He is an associate professor in Beijing University of Posts and Telecommunications, Beijing, China.

Qingqi Pei received his B.S., M.S. and Ph.D. degrees in Computer Science and Cryptography from Xidian University, in 1998, 2005 and 2008, respectively. He is now a Professor and member of the State Key Laboratory of Integrated Services Networks, also a Professional Member of ACM and Senior Member of IEEE. Senior Member of Chinese Institute of Electronics and China Computer Federation. His research interests focus on digital contents protection and wireless networks and security.
Yuan Liu received the B.S. degree in 2014 and the M.S. degree in 2018 from Henan Polytechnic University. She is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications. Her research interests include modern cryptography, network security.

Miaomiao Li is currently a M.S. candidate in Beijing University of Posts and Telecommunications. Her current research fields include network security, personal recommendation technology.