A trajectory privacy-preserving scheme based on query exchange in mobile social networks

Shaobo Zhang, Guojun Wang, Qin Liu & Jemal H. Abawajy
Your article is protected by copyright and all rights are held exclusively by Springer-Verlag GmbH Germany. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer’s website. The link must be accompanied by the following text: “The final publication is available at link.springer.com”.
A trajectory privacy-preserving scheme based on query exchange in mobile social networks

Shaobo Zhang1,2 · Guojun Wang3 · Qin Liu4 · Jemal H. Abawajy5

Published online: 12 June 2017
© Springer-Verlag GmbH Germany 2017

Abstract With the increase in the number of active location-based service (LBS) users, protecting the privacy of user trajectory has become a significant concern. In this paper, we propose a deviation-based query exchange (DQE) scheme that obfuscates the users’ query point to mitigate trajectory disclosure in mobile social networks (MSNs). The user finds a best matching user (BMU) in an MSN to exchange queries when a query request is issued. The DQE scheme can prevent the attacker from reconstructing the user’s trajectory by collecting data from the LBS server which records only the user’s ID and his BMUs’ locations (fake locations). By virtue of the private matching algorithm based on the matrix confusion, the DQE scheme allows LBS users to enjoy the service while preserving their privacy. In order to test the effectiveness and efficiency of the proposed scheme, we carried out extensive security and performance analyses under various conditions. The results of the experiments show that the proposed DQE scheme can protect users’ trajectory privacy effectively and reduce the overhead of the LBS server.

Keywords Location-based service · Trajectory privacy · Mobile social networks · Query exchange · Private matching

1 Introduction

Advances in wireless communication technologies and personal mobile devices with global positioning functionality and Internet connectivity have enabled the proliferation of mobile social networks (MSNs), which is a combination of online social networks and mobile location-based services (LBSs) (Ficco et al. 2016; Wang et al. 2016; Primault et al. 2016). In the MSNs, the users can utilize their handset location information to query the LBS provider for some point of interests (POIs) nearby (e.g., the nearest restaurant, hospital and shopping mall), or directly share information with nearby neighbors with common interests (Yi et al. 2016). In order to receive the desired service information, the user is required to report his/her current position to location service provider (LSP). For the mobile users, they have to report their positions continuously to retrieve LBS query results. As a user moves, a series of locations in the submitted queries form a user trajectory in the geographical space. Apart from giving LBS users access to many useful services, the collected user trajectories can be used in some real life applications, such as shopping advertisement, smart transportation and traffic navigation (Sui et al. 2017).
As trajectory information is a valuable commodity, the untrustworthy LSP may publish the user trajectories to a third party for analysis, which could pose serious risks to privacy breach. For example, an adversary can analyze the characteristics of the trajectory of a given user to discover the user’s workplace, home address, behavioral patterns and living habits. This fear has restricted the ultimate uptake of the LBSs since many potential users are concerned about their sensitive private information being leaked out (Xiao et al. 2017). As a result, privacy protection of user trajectory has recently become a major concern among the research community and industry.

There have been many user trajectory privacy-preserving techniques, such as the dummy trajectory method (Tang et al. 2016), the suppression method (Terrovitis and Mamoulis 2008) and the $k$-anonymity method (Niu et al. 2013). However, these approaches have some weaknesses. For example, the $k$-anonymity method always form some cloaking regions, when the user issues LBS requests in continuous query points. Although each point on the trajectory is protected based on the $k$-anonymity paradigm where at least $k - 1$ other users are included, the user trajectory can be easily disclosed under the following circumstances: (1) when the generated cloaking region is not big enough, the position information of each member can be easily revealed; (2) if the adversary compares the users of each cloaked region at a different time, it can also recognize the real user, and (3) if the adversary links these cloaked regions, the user’s entire trajectory will be exposed. As shown in Fig. 1a, the red line with the arrow represents the trajectory of a user (Alice), the stars on the trajectory represent the continuous query points at different times from $t_1$ to $t_5$, and the circles represent other users surrounding Alice. As Alice moves, she blurs each query point location in a cloaked region that satisfies the $k$-anonymity, where $k = 5$. However, the adversary can reconstruct the trajectory of Alice according to the sequence of the cloaked region. Therefore, the $k$-anonymity can hardly be guaranteed.

To overcome the defects of $k$-anonymity, we propose a deviation-based query exchange (DQE) scheme for trajectory privacy in the MSNs environment. When the user issues an LBS request, our scheme selects a best matching user (BMU) who possesses the maximum location deviation with the user in an MSN to exchange queries. As shown in Fig. 1b, users in an MSN need to query the POIs around their current location to the LSP. The red solid line with an arrow indicates the Alice’s real trajectory and direction, and she sends continuous queries on the trajectory at time points $t_1$–$t_5$. At $t_1$, she chooses a user (Charies) who possesses the maximum location deviation in an MSN to exchange queries. Then Alice’s query request is forwarded by Charies, and the LBS server records Charies’s ID and Alice’s location. At the same time, Charies’s query request is forwarded by Alice, and the LBS server records Alice’s ID and Charies’s location (fake location). At $t_2$, Alice selects David to exchange queries, and the LBS server records Alice’s ID and David’s location (fake location). At follow-up query points, Alice also exchanges queries with other users (David, Charies and Bob) from time points $t_3$–$t_5$ to obfuscate the continuous query points of her own. From these queries of Alice at time points $t_1$–$t_5$, the LBS server records the locations of Charies, David, David, Charies, and Bob, respectively, and the LSP can only infer the fake trajectory of Alice.

In the process of finding the BMU, our scheme utilizes the private matching algorithm based on matrix confusion
to achieve privacy preservation between the user and the
matching user, which means they will not reveal sensitive
information to each other. As the user sends the query to the
BMU who possesses the maximum location deviation every
time, the LBS server will not be able to deduce the relation-
ship between them. Furthermore, the LBS server records the
query points of the BMU, so it is impossible to infer the
user’s real trajectory through the query data. The following
summarizes the main contributions of this paper:

1. We propose a deviation-based query exchange scheme
for trajectory privacy protection in the MSNs environ-
ment. The user exchanges his/her LBS queries with the
BMUs in MSNs to prevent the LBS server from recon-
structing his/her real trajectory.
2. A method of finding a BMU with the maximum location
deviation is proposed to prevent the LBS server from deducing
the relationship between the user and his/her BMUs, and a private matching algorithm based on the
matrix confusion is used to achieve a secure and efficient
matching.
3. We perform extensive analysis of the proposed scheme to
study its effectiveness and efficiency. The results of the
experiments show that the proposed DQE scheme can
protect user trajectory privacy effectively and reduce the
overhead of the LBS server.

The rest of this paper is organized as follows. In Sect. 2,
we highlight the related works. In Sect. 3, we provide an
overview of our system architecture and definition. In Sect. 4,
we describe the DQE scheme in detail for trajectory privacy
protection. Then, the security analysis and performance anal-
ysis are provided in Sect. 5. Next, we run a set of simulations
to evaluate the performance of the DQE scheme in Sect. 6.
Finally, we conclude our work and present the future work
in Sect. 7.

2 Related work

In this section, we review related work on the privacy-
preserving techniques in LBSs. We discuss the two aspects of
privacy protection: location and trajectory (Chow and Mok-
bel 2011).

For location privacy protection, the main method is to
utilize k-anonymity to transform a location into a cloaked
region, making the individual indistinguishable from other
(k − 1) individuals (Niu et al. 2014; Sang et al. 2016). For
example, Chow (2006) first proposed a P2P cloaking algo-

rithm to generate a cloaking region where the mobile user
forms a group with his/her neighboring peers via single-hop
communication and/or multi-hop routing. Pan et al. (2012)
proposed an IClique cloaking algorithm which incremen-
tally maintains maximal cliques needed for location cloaking.
Li et al. (2016) introduced an incentive mechanism for k-
anonymity, and a user can gain his credit by providing assis-
tance to the others. However, this method has certain privacy
disclosure risk. Shokri et al. (2014) presented a scheme via
collaboration of the users. Interested users exchange context
information, it allows a querying user to answer LBS queries
from other users. In some cases, user interactions pose addi-
tional privacy risks. For this cloaking-based method, Jin et al.
(2014) characterized location injection attacks.

Trajectory privacy protection has received much attention
in recent years. Researchers have proposed some techniques,
such as dummy trajectories confusion, suppression-based
method, and trajectory k-anonymity. For dummy trajectories
confusion, the main challenge is how to generate dummy trajec-
tories (Lei et al. 2012). For example, Dai and Hua (2015)
proposed a method based on the segmented fake trajectory
under road network, which generated the fake position
for the sampling location of a real trajectory at a different
time, and generated the segmented fake trajectory in differ-
ent time intervals. For suppression-based method, Terrovitis
and Mamoulis (2008) proposed a method that iteratively sup-
presses selected locations from trajectories in order to satisfy
a privacy constraint. However, those trajectories are sup-
pressed too much, causing huge information loss. On the
whole, the methods of dummy trajectories confusion and
suppression-based are the simplest ways.

Trajectory k-anonymity has been widely applied to the
trajectory privacy protection. In trajectory k-anonymity, a user
will form a cloaked region which contains other (k − 1) users
who have the same probability to be linked to a specified iden-
tity; that way, the adversary cannot pinpoint the exact user
(Palanisamy and Liu 2015). For example, Chow and Mokbel
(2007) proposed a k-sharing region scheme, which supports
continuous queries. They believe that each location in a con-

tinuous query must contain other (k − 1) identical users, so
that the LBS server cannot identify the user’s location. Huo
et al. (2011) proposed a graph partition method to look for the
k-anonymity sets on the trajectory. In their follow-up work,
they even proposed a method that anonymizes the sensitive
stay points on trajectories to protect trajectory privacy (Huo
et al. 2012). In a recent study, Kim et al. (2013) proposed a
grid-based cloaking method in continuous LBSs. This cre-
ates a cloaking region by storing information and performing
operations in distributed systems. Hwang et al. (2014) pro-
posed a time-obfuscated technique, which forms the cloaking
region according to the user privacy profile and ambient con-

ditions, so the malicious LBS cannot reconstruct the user
trajectory. Schlegel et al. (2015) proposed a dynamic grid
scheme to support both k-nearest-neighbor and range queries
in continuous LBSs, which does not need a fully trusted third
party.
Table 1 Summary of notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSGU2B</td>
<td>The user sends query message to BMU</td>
</tr>
<tr>
<td>MSGB2S</td>
<td>The BMU forwards query message to LBS server</td>
</tr>
<tr>
<td>$E$</td>
<td>Asymmetric encryption function</td>
</tr>
<tr>
<td>$E_n$</td>
<td>Symmetric encryption function</td>
</tr>
<tr>
<td>PKS</td>
<td>Public key of the LBS server</td>
</tr>
<tr>
<td>SKS</td>
<td>Private key of the LBS server</td>
</tr>
<tr>
<td>KS</td>
<td>The key of symmetric encryption</td>
</tr>
<tr>
<td>R</td>
<td>The query range of the user</td>
</tr>
<tr>
<td>IDU</td>
<td>The identity of the user</td>
</tr>
<tr>
<td>IDBi</td>
<td>The identity of BMU at the $i$th query</td>
</tr>
<tr>
<td>Q</td>
<td>The content of query</td>
</tr>
<tr>
<td>MSG</td>
<td>The query results</td>
</tr>
<tr>
<td>MSGS2B</td>
<td>MSG from the LBS server to the BMU</td>
</tr>
<tr>
<td>MSGB2U</td>
<td>MSG from the BMU to the user</td>
</tr>
</tbody>
</table>

As mentioned above, the dummy trajectory method handles the whole trajectory, which will result in a large computation and storage overhead. The suppression technique of sensitive location may reduce the overhead, but the sensitive locations on trajectories are suppressed too much, causing a large information loss. In trajectory $k$-anonymity, a malicious LBS can get the user’s private information by observing the continuous queries. To overcome all of the defects, we propose a DQE scheme to obfuscate the user’s query points by BMU in MSNs. We believe this will achieve trajectory privacy protection, and it can reduce the overhead of LBS server’s query and communication.

3 The system model and definition

In this section, we first depict the DQE scheme for trajectory privacy protection in MSNs, and then we define some basic notions and provide the threat model. The notations used throughout this paper are summarized in Table 1.

3.1 System architecture

In a system of interest, the LBS provider (i.e., LBS server) advertises its location to all potential users. When LBS server receives the location information and service query from a user, it searches for the requested service data in database, and sends the results back to the user. It maintains a service database to store and update the service and associated data as well as a public and a private key. It is considered an honest but curious entity, which may leak the user’s sensitive information to an adversary. The adversary can use the trajectory data to link the user’s data reports with sensitive locations.

LBS users have mobile handsets which are capable of computation, storage and wireless communication as well as global positioning functionality (e.g., GPS). The architecture of DQE scheme is shown in Fig. 2. Users are assumed to be members of the MSN. When a user issues a series of queries to the LSP in continuous LBSs, a trajectory is formed. The stars on the trajectory represent a sequence of point queries issued by the user at different times from $T_1$ to $T_3$, and the circle represents the range covered by the MSN. The BMU is a user who matches certain attributes of a specified user in an MSN. In our scheme, the BMU has the maximum location deviation and has trajectory segments different from the user in an MSN.

In our scheme, the communication channels in MSNs are assumed to be secure when a user is trying to find a BMU. The existing security schemes (e.g., mutual authentication and key agreement) and conventional solutions (e.g., cryptography and hashing) can be used to protect the privacy and integrity of the information through the network (Wang et al. 2013; Fu et al. 2015; Kumari et al. 2016).

Definition 1 (Best matching) The best matching is the maximum location deviation and has the difference of trajectory segments between the user and the BMU in an MSN.

Definition 2 (Location) A location $L$ is a two-tuple $(x, y)$ which represents the latitude and longitude of the location.

Definition 3 (Trajectory) For a moving object $Q$, its trajectory $T$ is a set of discrete locations at sampling time, which can be:

$$T = \{ID_Q, (x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)\},$$

$$t_1 < t_2 < \cdots < t_n$$
Here ID_Q represents the trajectory identity of moving object Q, \((x_i, y_i, t_i)\) represents the sample location of moving object Q at time \(t_i\) on the trajectory.

Our goal is to guarantee the privacy of the trajectory data of LBS users and enable the system to provide high-quality service. If the LBS server is compromised, the adversary is able to collect all LBS data, which is a sequence of continuous LBS queries issued by the user. Therefore, the adversary is very likely to infer the user’s trajectory successfully from the data and reveal sensitive information.

### 3.2 The attack model

We assume that the main goal of the adversary is to identify the user’s real trajectory associated with the identity. A common adversary can be an entity that eavesdrops on a wireless channel between BMU and LBS server, or an attacker who has compromised the LSP. Based on the sensitive information which an adversary can get, we consider both the weak and strong adversary attack models described in Gao et al. (2013) and Zhang et al. (2015).

**Weak adversary attack model** The weak adversary has little knowledge about the user. It can only an adversary that can wiretap the insecure wireless channel. Eavesdroppers are usually local, short-term and passive because of their status features and limited resources. He tries to infer some sensitive information of user from the eavesdropped information, such as sensitive locations of the user, user’s identity and interest.

**Strong adversary attack model** The strong adversary has more power than the weak adversary. At the worst case, it can compromise the LSP and leak the sensitive information for profit. The LBS server is capable of being a global, long-term, and active observer, and it manages all the queries for users’ server and records all the behaviors of a particular user.

### 4 The DQE trajectory privacy protection

In this section, we present the DQE scheme for trajectory privacy protection in LBSs. First, we describe the concrete working processes of this scheme, and then we discuss in detail the private matching process. The system consists of the user, the BMU, and the LBS server.

#### 4.1 The DQE scheme

Our basic idea is to introduce an MSN and find a BMU in it to exchange queries with the user at different query points, then they send the queries to LBS server respectively. The working process of the DQE scheme mainly consists of five steps, as shown in Fig. 3 and is described below.

**Step 1 (Private matching)** The user runs Algorithm 3 (details in Sect. 4.2) to find a BMU. The process of finding the BMU is illustrated in Fig. 4. In the example, Grace is the user, who initiates the service request by sending a query. This creates an MSN that consisting of seven users. In this MSN, Frank has the maximum location deviation and the largest trajectory difference with Grace. Thus he is selected as her BMU for query exchange.

Then Grace generates and sends Frank a query message \(MSG_{U2B}\). We formalize the request message by user
$U(\text{MSG}_{U2B})$ to the selected BMU as follows:

$$\text{MSG}_{U2B} = \{E_{PK_U}(ID_U), E_{PK_S}(Q, R, K_S)\}$$  \hfill (2)

where $U$ is the user and $ID_U$ is his/her identity. The query request consists of the content ($Q$), the query range ($R$) and a key for symmetric encryption between the LBS server and the user ($K_S$). The query request is encrypted using the LBS server public key (PK$_S$). The user ($ID_U$) is encrypted using the BMU public key (PK$_B$). It is appended to the encrypted query request and then sent to the selected BMU.

**Step 2 (ID transformation)** The BMU extracts the user’s real ID ($ID_U$) from MSG$_{U2B}$, transforms the $ID_U$ into the BMU’s pseudo-ID ($ID_{Bi}$), and stores both $ID_U$ and $ID_{Bi}$ in the file list. Then the BMU generates a query message composed of the query request from $ID_U$ and BMU’s pseudo-ID ($ID_{Bi}$) as follows:

$$\text{MSG}_{B2S} = \{E_{PK_S}(ID_{Bi}), E_{PK_S}(Q, R, K_S)\}$$  \hfill (3)

Finally, the BMU sends MSG$_{B2S}$ to the LBS server. Note that the BMU does not know the content of the query request from the user as it is encrypted with the public key of the LBS server (PK$_S$).

**Step 3 (Searching POIs)** When the LBS server receives MSG$_{B2S}$ from the MBU, it first decrypts the message and extracts the query content, the query range and the pseudo-ID. Then, the LBS server searches the POIs and obtains the query results (MSG) from the POIs. Finally, it encrypts the MSG using the symmetric encryption key ($K_S$) and sends it to the BMU.

$$\text{MSG}_{S2B} = \{E_{PK_U}(ID_{Bi}), \text{En}_{K_S}(\text{MSG})\}$$  \hfill (4)

**Step 4 (ID recovery)** When BMU receives MSG$_{S2B}$ from the LBS server, it first recovers the identity of the user $ID_U$ from the file list, and extracts the encrypted query results ($E_{K_S}(\text{MSG})$) from MSG$_{S2B}$. Then it will encrypt the $E_{K_S}(\text{MSG})$ with the public key of the user (PK$_U$). Finally, the BMU exchanges result messages with the user, and sends MSG$_{B2U}$ to the user.

$$\text{MSG}_{B2U} = E_{PK_U}(\text{En}_{K_S}(\text{MSG}))$$  \hfill (5)

**Step 5 (Accurate results)** When the user receives MSG$_{B2U}$ from the BMU, it extracts the MSG using its private key and symmetric key and then gets the accurate results. In the query exchange process, the BMU also executes the steps 1–4 to obtain the accurate results.

### 4.2 The private matching technique

In this section, we present a matrix confusion technique to protect the user matching privacy. First, we randomly select $k$ users in an MSN as candidates. This is done to avoid generating a large number of messages in attribute matching. Second, we generate the matching value between the initiator and the candidates. This is done by selecting a group of points from the trajectory segments of the user within the time interval $\Delta t$ ($3 \text{ ms} \leq \Delta t \leq 10 \text{ ms}$) when the user sends the matching query. Then, we calculate the matching values of each pair of points between the initiator and candidates in an MSN.

#### 4.2.1 The matrix confusion

The users in MSNs have three common attributes: the position coordinate values $x$ and $y$, and the movement direction $\theta$. The $x$ and $y$ denote the $X$ and $Y$ axis coordinate values of the user’s location respectively, and $\theta$ denotes the current movement direction of the user, $\theta \in [0, 360]$. Using these three common attributes and their levels, we create the attribute matrix $M_{n \times 3}$, where $m_{ij} \in M_{n \times 3}$, $m_{ij} = 0$ or $1$, $1 \leq i \leq n$, $1 \leq j \leq 3$.

$$M_{n \times 3} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ \vdots & \vdots & \vdots \\ m_{n1} & m_{n2} & m_{n3} \end{bmatrix}$$

where the column vectors represent the common attributes of $x$, $y$, and $\theta$, and the row vectors indicate the levels of attribute. For example, we take every angle of $360/n$ degrees as one unit angle and then, we select a corresponding integer $i$, $i \in [1, n]$, which indicates the level of the attribute $\theta$. If the $i$ is a user’s level of the third attribute in $M_{n \times 3}$, he can set the $m_{ij} = 1$ and $m_{j3} = 0$ where $j \neq i$, $m_{j3} \in M_{n \times 3}$.

The users in MSNs can be defined as the initiator (i.e., the user) and the candidate, and each user has an personal attribute matrix $M_{n \times 3}$. To find the BMU in private, we first encrypt the initiator’s matrix $I_{n \times 3}$ by hiding personal information via matrix confusion. The process is shown in Algorithm 1.

The inputs of the algorithm are the initiator’s attribute matrix ($I_{n \times 3}$) and two large prime numbers $\alpha$ and $\beta$, where $|\alpha| = 256$, $\beta > 4n^2\alpha^2$. The output of the algorithm is the confusion matrix ($I_{n \times 3}$) that was created by randomly generating two matrices $P_{n \times 3}$, $R_{n \times 3}$ and encrypted by the keys $k_1$, $k_2$ and $k_3$. The initiator gets the confusion matrix $I_{n \times 3}^*$, broadcasts his/her matching discovery request and sends the $I_{n \times 3}^*$ to other $k$ candidates in the MSN. Since the candidate does
not have any knowledge of confusion matrix, he/she cannot
deduce the real attribute information about the initiator.

**Algorithm 1** The process of matrix confusion

**Input:** $I_{n \times 3}$ (Initiator’s attribute matrix), $\alpha, \beta$ (Large prime numbers)

**Output:** $I^*_{n \times 3}$ (The confusion matrix)

1. Randomly generate two matrices $P_{n \times 3}, R_{n \times 3}, \forall p_{ij} \in P_{n \times 3}, r_{ij} \in R_{n \times 3}, \sum_{i=1}^n \sum_{j=1}^n p_{ij} \leq (\alpha-3n), |r_{ij}| \beta \approx 1024, 1 \leq i \leq n, 1 \leq j \leq 3;

2. $\forall i_{ij} \in I_{n \times 3}, \forall j_{ij} \in I^*_{n \times 3}, k_i \in K$,
   - the initiator confuses the matrix $I_{n \times 3}$ by the following operations:
   - for $i \leftarrow 1$ to $n$
     - $k_i = 0$;
     - for $j \leftarrow 1$ to $3$
       - if $i_{ij} = 1$ then
         - $i^*_{ij} = \alpha + p_{ij} + r_{ij} \beta$;
       - else
         - $i^*_{ij} = p_{ij} + r_{ij} \beta$;
     - end if
     - $k_i = k_i + (r_{ij} \beta - p_{ij})$;
   - end for
   - end for
3. **return** $I^*_{n \times 3}$

4.2.2 Calculating the similarity value

The weight matrix $W_{n \times n}$ is used to indicate different attention
degrees for the attributes (Luo et al. 2017). The greater the
similarity attribute value the bigger the weight will be. This is
defined in Eq. (6).

$$(W)_{n \times n} = (w_{ij})_{n \times n} = \begin{cases} n + i^2, & i = j \\ n + j - i, & i - j > 0 \\ n - j + i, & i - j < 0 \end{cases}$$ (6)

When the candidates receive the request from the initiator,
it can execute a matrix multiplication operator between the
candidate matrix $C_{n \times 3}$ and the confusion matrix $I^*_{n \times 3}$. Then,
the candidates send computation result $A_{n \times n}$ to the
initiator. The initiator does further transformation to get the $B_{n \times n}$
and performs the similarity calculation operation with the
Corresponding weights. We can then get the similarity value
$\varphi$, which is the weighted average similarity value between
the initiator and candidate. A bigger $\varphi$ indicates that the two
points are near, and the movement direction $\theta$ is similar. The
calculation of similarity value is shown in Algorithm 2.

In order to resist continuous tracing attacks, we choose a
BMU that is farthest away from the initiator and the smallest
of the similarity value. For example, the similarity value
between initiator and Alice is 9, and between initiator and
Bob is 5. Obviously, we should choose Bob as the BMU
candidate.

**Algorithm 2** The calculation of similarity value

**Input:** $I^*_{n \times 3}$ (Initiator’s confusion matrix), $C_{n \times 3}$ (Candidate matrix),
$W_{n \times n}$ (Weight matrix)

**Output:** $\varphi$ (The similarity value)

1. for all $i_{ij} \in I^*_{n \times 3}, c_{ij} \in C_{n \times 3}$ do
2. $I^*_{n \times 3}$ was sent to the candidates to compute $A_{n \times n} = (a_{ij})_{n \times n} = I^*_{n \times 3} \times C_{n \times 3}$ by the following operations:
3. for $i \leftarrow 1$ to $n$
   - for $j \leftarrow 1$ to $n$
     - $a_{ij} = a_{ij} + \alpha \times i_{ij}^*$;
   - end for
4. end for
5. $A_{n \times n}$ was sent to the initiator, then the initiator makes a further transformation to get $B_{n \times n} = (b_{ij})_{n \times n}$, where $b_{ij} = (a_{ij} - (b_{ij} \bmod \alpha^2))/\alpha^2, b_{ij} = (a_{ij} + k_i) \bmod \beta$, for $a_{ij} \in A_{n \times n}, k_i \in K$;
6. The initiator considers the corresponding weights and computes $D_{n \times n} = (d_{ij})_{n \times n} = B_{n \times n} \times W_{n \times n}$;
7. Calculate the similarity value: $\varphi = \sum_{i=1}^n \sum_{j=1}^n d_{ij}$;
8. **return** $\varphi$

4.2.3 Finding the BMU

An illustration of the matching process is shown in Fig. 5.
There are two trajectory segments $U$ and $S$ formed by the
user (initiator) and the Alice (the $i$th candidate) within the
time interval $\Delta t$. Each trajectory segment includes a group
of points that indicate $m$ footprints of the initiator and the
candidate from $t_1$ to $t_m$. We can calculate the similarity value
$\varphi_{ij}$ of two points at $t_i$, where $1 \leq i \leq k, 1 \leq j \leq m$.

![Fig. 5 The process of the matching](image-url)
After getting the similarity value \( \varphi_{ij} \) of each pair of points, we can calculate the average similarity value \( E(\varphi_i) \) and the variance \( D(\varphi_{ij}) \) of each group of points which contains \( m \) similarity values between the user and the candidate in the MSN. The smaller the average similarity value, the greater the location deviation degree, and the variance denotes the difference of the trajectory segments. The variance is calculated as follows:

\[
D(\varphi_i) = E \left\{ \left[ \varphi_{ij} - E(\varphi_i) \right]^2 \right\}, 1 \leq i \leq k, 1 \leq j \leq m
\] (7)

where \( \varphi_{ij} \) denotes the similarity value of the \( j \)th point between the \( i \)th candidate and the user, and \( E(\varphi_i) \) denotes the average similarity value of the \( m \) points. The calculation of the best matching value is shown in Algorithm 3, and the candidate with the best matching value is the BMU.

**Algorithm 3** The calculation of best matching value

**Input:** The initiator’s matrix \( I_{m \times 3} \) and the candidate’s attribute matrix \( (C_{m \times 3}), 1 \leq i \leq km \)

**Output:** The best matching value \( \psi \)

1. for \( i \leftarrow 1 \) to \( k \) do
2. \( \varphi_i = 0 \);
3. for \( j \leftarrow 1 \) to \( m \) do
4. \( \varphi_i = \varphi_i + \varphi_{ij} \);
5. end for
6. \( E(\varphi_i) = \varphi_i / m \);
7. \( D(\varphi_i) = E(\left[ \varphi_{ij} - E(\varphi_i) \right]^2) \);
8. if \( i = 1 \) then
9. \( \psi = E(\varphi_i) \);
10. end if
11. if \( E(\varphi_i) < \psi \) and \( D(\varphi_i) \neq 0 \) then
12. \( \psi = E(\varphi_i) \);
13. end if
14. end for
15. return \( \psi \)

### 5 Analysis

#### 5.1 Security analysis

Our security analysis will focus on how the DQE scheme achieves user trajectory privacy. The threat model has been discussed in Sect. 3.2 Here, we analyze the security of the proposed scheme against eavesdropping attacks and dishonest LSP.

**5.1.1 Resistance to eavesdropping attacks**

The attackers can monitor and eavesdrop on communication processes between the user and the LBS server across wireless channels. Our DQE scheme uses cryptography techniques to deal with eavesdropping attacks. All of the messages transmitted in the wireless channel are secured by asymmetric and symmetric encryptions.

When the user sends the query message to the LBS server, the message will be encrypted by asymmetrical algorithm \( E \). The user first encrypts the query content, the query range as well as the private key into \( E_{PK_S}(Q, R, K_S) \) by using public key \((PK_S)\) of the LBS server, and \( ID_U \) is encrypted by the public key \((PK_B)\) of the BMU, and then sends \( MSG_{UB2B} = \{E_{PK_B}(ID_U), E_{PK_S}(Q, R, K_S)\} \) to the BMU. In this process, the attackers do not have the private key of the BMU and the LBS server, so they cannot get any information from the message \( MSG_{UB2B} \). Similarly, when the BMU forwards \( MSG_{B2S} = \{E_{PK_S}(ID_Bi), E_{PK_S}(Q, R, K_S)\} \) to the LBS server, the attackers cannot get any information from the message \( MSG_{B2S} \). Therefore, the sensitive information will not be disclosed.

When returning query results, the \( MSG_{SB2B} \) and the \( MSG_{B2U} \) include \( E_{PK_S}(ID_Bi) \) and \( En_{K_S}(MSG) \). The \( ID_Bi \) is encrypted by asymmetrical algorithm \( E \) with the public key \((PK_B)\) of the BMU, and the query results \((MSG)\) is encrypted by symmetrical algorithm \( En \) with the key \( K_S \) of the user. Attackers do not have access to these keys, so the probability of them getting the useful information is negligible.

From the above analysis, our DQE scheme can resist the attack of eavesdropper, and attackers cannot gain the real identity of the user, the accurate query location, and the query content.

#### 5.1.2 Privacy against LSP

The LSP manages all of the query information from all users and acts as an honest but curious attacker to infer sensitive information from these data, including the real trajectory of the user.

In our scheme, we exchange queries between the user and the BMU, then send it to the LSP. In this process, the identity of the user \( ID_U \) is replaced by BMU’s \( ID_Bi \) in the \( i \)th query, and the record of query information store in the LSP is linked to the BMU’s \( ID_Bi \). Since the BMU changes dynamically, i.e., it is different at different query points, the LSP will not be able to deduce the relationship between them and it will not be able to recognize the user’s real trajectory from the arbitrary BMU’s \( ID_Bi \). Therefore, the probability of the LSP (or attackers of the LSP) being able to infer the user’s real identity or its trajectory is negligible.

#### 5.2 Performance analysis

We will analyze the performance of the proposed DQE scheme in terms of the computational cost and the communication cost.
5.2.1 Computational cost

We first consider the computational overhead on the user, which is mainly on finding a BMU in an MSN at each query point. We select \( k \) users in an MSN and utilize a private matching technique based on the matrix confusion to achieve matching. Its computation complexity is \( O(km) \) (Zhu et al. 2014), where \( m \) is the number of points on the trajectory segment.

The computational overhead of the BMU is mainly due to the need to calculate the average similarity value (i.e., \( O(m) \), where \( m \) is the number of points on the trajectory segment) and exchange query with the user. In the latter, one needs to send the query request to the LBS server and forward the query results to the user. Its computation complexity is \( O(C) \), where \( C \) is a constant. So, the computational cost on the BMU is \( O(m) \).

The computational overhead on the LBS server is mainly on searching the databases for the user’s POIs. The computation cost of this search in Vu et al. (2012) is \( O(n) \), where \( n \) is the number of the POIs.

5.2.2 Communication cost

We first consider the communication overhead required to find a BMU in an MSN. We utilize a private matching technique to find the BMU in the \( k \) users for matching. Its communication cost is \( O(km) \) (Zhu et al. 2014), where \( m \) is the number of points on the trajectory segment.

Then we consider the communication cost between the user and the BMU. The query message of the user to the BMU is MSGU2B in (2), whose size is a constant. The query results message forwarded to the user by the BMU is MSGB2U in (5). Its size varies with the number of POIs, \( n \). So the communication cost between the user and the BMU is \( O(n) \).

Finally, we consider the communication cost between the BMU and the LBS server. The query message sent to the LBS server by the BMU is MSGB2S in (3), whose size is a constant. The query result message returned to the BMU is MSGS2B in (4). Its size varies with the number of POIs, \( n \), so the communication cost between the BMU and the LBS server is \( O(n) \).

In Table 2, we summarize the computation and the communication overheads at the user, the BMU, and the LBS server levels.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Computation cost</th>
<th>Communication cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>The user</td>
<td>( O(km) )</td>
<td>( O(km) )</td>
</tr>
<tr>
<td>The BMU</td>
<td>( O(m) )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>The LBS server</td>
<td>( O(n) )</td>
<td>( O(n) )</td>
</tr>
</tbody>
</table>

We first describe the experiment setup in Sect. 6.1, then we evaluate the performance in Sects. 6.2 and 6.3.

6 Evaluation

In this section, the effectiveness and efficiency of our proposed DQE scheme are experimentally evaluated under various system settings. We conducted our experiments focusing on the overhead of the user side and the LBS server.

6.2 Effectiveness analysis

We will analyze the effectiveness of our DQE scheme from the overhead of the user side and the LBS server.

6.2.1 The overhead of the user side

On the user side, the overhead is mainly on the private matching technique to find the BMU, which is affected by the number of users \( k \) in an MSN and the number of points on the trajectory segment \( m \) and the level of attribute \( n \). For \( k = 10 \), \( 20 \) and \( 30 \), we will evaluate the effect of \( m \) and \( n \) on the performance of DQE scheme.

As shown in Fig. 6, under the same value of \( k \) and \( n = 3 \), the processing time and the communication cost on the user side increase with the increase of \( m \). For example, when \( k = 10 \), the processing time on the user side gradually increased from 1.66 to 5.3 s, and the communication cost gradually increased from 56 to 230 kB as \( m \) gradually increased from 3 to 10. This indicates that the processing time and the communication cost on the user side are linear.
Table 3  Experimental parameter settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( j )</td>
<td>Number of user’s attributes</td>
<td>3</td>
</tr>
<tr>
<td>POIs</td>
<td>Number of point of interests</td>
<td>2000, 4000, 6000</td>
</tr>
<tr>
<td>( m )</td>
<td>Number of points on the trajectory segment</td>
<td>3–10</td>
</tr>
<tr>
<td>( n )</td>
<td>The level of the attribute</td>
<td>2–9</td>
</tr>
</tbody>
</table>

Fig. 6  The effect of the number of points \( m \). \( a \) Processing time versus \( m, n = 3 \), \( b \) communication cost versus \( m, n = 3 \)

Fig. 7  The effect of the attribute level \( n \). \( a \) Processing time versus \( n, m = 3 \), \( b \) communication cost versus \( n, m = 3 \)

to the number of the points on the trajectory segment. When we increase the number of users \( k \) which we randomly select, the user needs to match more users in an MSN, which then incurs more time and greater overhead on finding the BMU. Therefore, the processing time and communication cost on finding the BMU increase with the value of \( k \) or \( m \).

As shown in Fig. 7, under the same value of \( k \) and \( m = 3 \), the overhead on the user side increases with \( n \). For example, when \( k = 10 \), the processing time on the user side gradually increases from 1.21 to 4.3 s, and the communication cost gradually increases from 30 to 372 kB as \( n \) gradually increases from 2 to 9. Under the same value of \( n \), as the value of \( k \) grows, the processing time will increase, and so does the communication overhead. Overall, the processing time and communication cost of finding the BMU increase with the value of \( k \) or \( n \).

6.2.2 The overhead of the LBS server

The computation cost on the LBS server mainly focuses on searching POIs. As shown in Fig. 8a. For \( \text{POIs} = 2000, 4000 \) and \( 6000 \), the processing time on the LBS server is 82, 108 and 132 ms, respectively. The processing time does not change with \( k \), but it increases from 82 to 132 ms with \( \text{POIs} \). It is clear that the computational overhead of the LBS server is independent of \( k \). However, as the number of the POIs becomes larger, longer processing time will be needed.

The communication cost of the LBS server includes \( \text{MSG}_{2B} \) and \( \text{MSG}_{S2B} \) (as discussed in Sect. 4.1). The former is an encrypted message, which is a constant, while the latter is a set of POIs, the size of which varies. As shown in Fig. 8b, for \( \text{POIs} = 2000, 4000 \) and \( 6000 \), the communication overhead on the LBS server were 2.8, 4.0 and 5.2 kB, respec-
6.3 Comparison

We compared the effectiveness of our DQE scheme with the 3PLUS (Niu et al. 2013) and the P2P (Chow 2006) anonymity methods in terms of the overhead on the user side and the LBS server. In the following simulations, we compared the three methods by changing the value of $k$ from 10 to 100, when $n = 3$, $m = 3$, POIs = 4000, and the other parameters remained unchanged.

For the overhead on the user side, we mainly compared the processing time. Our DQE scheme finds the BMU by the private matching technique based on the matrix confusion in an MSN, whereas the anonymity method generates the $k$-anonymity spatial region.

As shown in Fig. 9, the processing time of all three methods increased with $k$. This is because the 3PLUS and P2P methods need to generate the spatial cloaking region that included $k$ users, and the DQE scheme needs to match $k$ users in an MSN. However, as the number of users $k$ increased, the processing time of 3PLUS and P2P methods increased rapidly, while the DQE scheme’s increase was gradual. This is an advantage on the user side.

As for the overhead on the LBS server, we compared the processing time and the communication cost of the three
methods with the different values of $k$. The processing time mainly depends on the search for POIs.

As shown in Fig. 10a, the processing time of the 3PLUS and P2P methods gradually increased with $k$, whereas the DQE scheme remained unchanged. It is clear that the higher the $k$, the larger the cloaking region the 3PLUS and P2P methods need to generate. Consequently, the LBS server needed to query more POIs in the cloaking region. However, the DQE scheme is independent of $k$ when it comes to the processing time on the LBS server.

Figure 10b shows that the communication cost of the 3PLUS and P2P methods gradually increased with $k$, whereas the DQE scheme remained unchanged. This is because that the 3PLUS and P2P methods vary with the anonymity degree $k$. The DQE scheme, on the other hand, is independent of $k$ when it comes to the communication overhead on the LBS server.

7 Conclusion

In this paper, we proposed an efficient deviation-based query exchange scheme, DQE, to protect trajectory privacy in MSNs. Our DQE scheme is based on the discovery of the BMUs in MSNs to exchange queries with the user. It can obfuscate the user’s query points, so that the LSP cannot reconstruct the user's real trajectory through continuous queries. Security and performance analysis show that our scheme can preserve privacy more effectively than the anonymity scheme.

However, the users in MSNs just abide by mutual forwarding mechanism, which can not guarantee the quality of information forwarding. If the BMU in an MSN can not immediately forward the request message of the user, it will lead to the delay of the user query. Therefore, as part of our future work, we intend to introduce an incentive mechanism for users to forward the query information in MSNs in order to improve the service quality.

Acknowledgements This work is supported in part by the National Natural Science Foundation of China under Grant Nos. 61632009, 61472451 and 61402161, the High Level Talents Program of Higher Education in Guangdong Province under Grant No. 2016ZJ01, the Hunan Provincial Education Department of China under Grant No. 2015C0589, the Hunan Provincial Natural Science Foundation of China under Grant No. 2015JJ3046, the Fundamental Research Funds for the Central Universities of Central South University under Grant No. 2016zzts058.

Compliance with ethical standards

Conflict of interest The authors declare that there are no conflicts of interest.

Human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

References


Chow CY, Mokbel MF (2011) Trajectory privacy in location-based services and data publication. ACM, New york


Springer


