Research Article

A New Privacy-Preserving Scheme for Continuous Query in Location-Based Social Networking Services

Eric Ke Wang¹ and Yunming Ye¹,²

¹ Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China
² Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China

Correspondence should be addressed to Eric Ke Wang; wk_hit@hit.edu.cn

Received 24 November 2013; Accepted 11 March 2014; Published 4 May 2014

Academic Editor: Hoon Ko

Copyright © 2014 E. K. Wang and Y. Ye. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With continuous queries that are used widely in location based mobile social networking services, how to protect the location privacy effectively for continuous query has been a hot topic for researchers. In this paper, we analyze the existing location privacy protection systems and algorithms for location based services; considering their disadvantages of slow responding time and high anonymization costs, we propose a new enhanced greedy cloaking algorithm which predicts a cloaking area at the initial query to be the cloaking region in the whole query lifetime by the comprehensive computation of privacy monitor, quality monitor, and dynamic adjuster. Privacy monitor and quality monitor charge the privacy protection level and service quality degree respectively; dynamic adjuster can adjust the cycle center point dynamically. We employ cycle as cloaking region form which can effectively alleviate the computation overhead. And we compare it with the earlier algorithm on three aspects. The experimental result shows that the enhanced greedy cloaking algorithm is better than the original greedy algorithm on average responding time or anonymization cost.

1. Introduction

With the rapid development of location based mobile social networking services (LBMSS) [1, 2], location information has been a key factor for improving the quality of location based mobile social networking services. However, when users enjoy the convenience and efficiency of LBMSS, the location privacy leakage problems have become more and more serious than before. For example, Loopt [3] is a popular mobile social networking application which can create a map on the phone that will locate users’ friend’s residences and where they have visited; when users want to maintain the social talks with their nearest friends, they have to contentiously query the friends’ locations. The adversary can infer the privacy of these people from the query content history and the people’s location history. Although there are some mature measures such as K-anonymity scheme [4–6] to solve it, most of them resolve the snapshot query that the content of any two consecutive queries from one user is uncorrelated. But in the mobile social networks users often need to repeat queries on a certain content during a time. For instance, in the mobile social navigation application WAZE [7], the road traffic information is gathered by reports from all users based on their current positions. When users report the road traffic situation based on their current locations, they probably need to continuously query the latitude and longitude of current location to report the road situation on continuous time. We call this kind of queries as continuous query. Directly applying the privacy protection algorithms for snapshot query to continuous query would cause some problems; the bad guys can infer the users’ exact positions according to some contextual knowledge [8, 9]. As an example, a common traditional location protection method for snapshot query is that replacing users’ location position with a space region which consists of k users who request queries in the same time period. However, for continuous query, if the above method is employed, there will be many consecutive space regions at different time which probably have intersections.
with each other, and the intersection part may be used to infer the users’ location. Later on, some researchers [10–12] proposed schemes that a region is generated at first query to be the cloaked region in the whole life cycle of continuous queries. Thus there is no need to worry about intersection inference problems. Yet, how to select the first cloaked region and how to prevent the mobile users of continuous queries from converging to one point or be too dispersive have been a new research problem to be solved.

2. Background

Traditional privacy protection algorithms such as $k$-anonymity (replace one query user’s location with a region consisting of at least $k$ users) are mainly for snapshot query that a user issues one query on each interest point. Each query is uncorrelated with others. But in physical world, continuous query is more common than snapshot query such as GPS navigation for the users’ nearest friend in a time period. When users who request queries are moving, the adversaries can collect the different cloaked regions to infer the users’ exact location or movement route.

Here we give the details of some typical location privacy inference problems.

2.1. Cloaked Regions Intersection Problem. Cloaked regions intersection problem is that the adversary may infer the user’s exact location without having the user’s any contextual knowledge. The adversary gets the consecutive cloaked regions to calculate the intersection part from these regions to understand the exact location of the user.

When a user send a continuous query to the location anonymizer (A server who charging anonymizing users queries is called as an anonymizer), the anonymizer will generate consecutive different cloaked regions for one query. The adversary can explore the cloaked regions to find out the user’s location. An example is presented in Figure 1.

In Figure 1, mobile user A issues a 3-anonymity continuous query, which generates three different cloaked regions on three consecutive time points. Figure 1(a) shows the cloaked region a on $t_1$, which includes users A, D, and E; because users are always mobile, on $t_{i+1}$, cloaked region b is different from a as it includes users A, E, and F, while D is not inside the cloaked region at that moment; on $t_{i+2}$, cloaked region c is generated where it includes users A, B, and H; if adversaries have the abilities to intercept the data of the three cloaked regions, he only need to calculate the intersection part and can easily find out that the query is from user A.

2.2. Privacy Inference Problems Based on Contextual Knowledge. Commonly location data in location based mobile social networking services consists of users’ coordinates {longitude, latitude}, velocity, and direction. We called these data users’ contextual information. If one or some items are known by the adversary, he can easily infer the users’ sensitive privacy. Figure 2 shows how the privacy is inferred when mobile users’ direction and velocity are known by the adversary.

As shown in Figure 2(a), the black solid point is a user issuing queries; gray solid points are other mobile users. Three-anonymity cloaked region is generated like the black rectangle at $t_1$. Suppose the movement direction of the query user is parallel with coordinate $x$, the movement speed is a constant $V$, and thus the adversary can calculate the possible cloaked region according to $V$ and direction. The adversary uses the cloaked region of $t_1$ to move to left and right by $V$, on the gray area is the intersection part of inferred cloaked region and real cloaked region, which only includes one user. Thus the query user is exposed. In Figure 2(b), if the velocity of the query user is a certain value in a range $(V_{\text{min}}, V_{\text{max}})$, the attackers are unsure of the exact velocity of the user; however, inference process can be similar. The attackers use $V_{\text{min}}$ and $V_{\text{max}}$ to calculate the possible cloaked regions on the left and right side. At $t_2$, the distance of the left boundary movement is $V_{\text{min}}(t_2 - t_1)$, while the right boundary is $V_{\text{max}}(t_2 - t_1);$ thus
the query user is still exposed. The above attacking measures are both based on the knowledge of direction of the users. However, even without the direction information, privacy can be still leakage.

As shown in Figure 3(a), the direction of mobile user is unknown but a range of angle \((\theta_{\min}, \theta_{\max})\), velocity is constant \(V\). Based on the knowledge, the adversary can calculate a possible cloaked region (an irregular close region shown as dotted line area in Figure 3(b)). This means that when direction and velocity are all uncertain, the early attacking measure still works.

According to the above analysis, we can find that the capability to protect the privacy of continuous query is very limited. The adversary can analyze the users’ independent cloaked regions or some other background knowledge, and then users’ privacy may be inferred and exposed. Therefore, we need to research on new advanced privacy preserving scheme for continuous query of location based social networking services.

For the privacy protection of continuous query, some researchers have proposed some schemes. Chow et al. [11,12] proposed a scheme that the users included in the initial cloaked area will last in the whole query life cycle; in other words, it is that the cloaked region selected at the first should also works at the end of the continuous queries. As shown in Figure 4.

User A sends a 3-anonymity query, at initial time \(t_1\), \(\{A, E, D\}\) are the users waiting to be anonymized in the generated cloaked region; at \(t_2\), the cloaked region maintains \(\{A, E, D\}\) as cloaking users, and adjusts the size of cloaked region according to the moving distances of the three users from time \(t_1\) to \(t_2\); at \(t_3\), the same process is executed; thus the adversary cannot get the real user by intersection of the consecutive cloaked regions. However, its biggest drawback is that the users who decided to be in a region are close at the initial time, but when the time goes, users move in different direction and speed, and thus it can result in disperse distribution at the next time, the size of cloaked region may be too big to reduce the quality of location based social networking services, or the users may gather to one same position at the next time, which can also expose the users’ privacy. To solve this problem, [13] gives a \(\delta_p\)-privacy model and \(\delta_q\)-quality model that select the mobile users in the initial cloaked region by considering the possible movement area of mobile users in the valid query life cycle. \(\delta_p\)-privacy model mainly limits the length and width of the cloaked region to guarantee that the users of continuous query do not gather to one point; \(\delta_q\)-quality model can monitor and adjust the torsion level to guarantee the service quality. But the two models have some disadvantages; one is that solving the rectangle region needs to calculate both parameters (length and width), which cost the computation of server; secondly the system needs to make real-time update the length and width by monitoring users’ movement and predicting the next positions of the mobile users, which aggravate the burden of servers. To solve the second problem, [14] gives a
method by cloaking the enough factors (location coordinates, velocity, and direction) of mobile users; thus the possible cloaked regions calculated by the adversaries are totally inside the real cloaked regions. As shown in Figure 5. However, the accuracy of algorithm depends on the estimation of mobile users’ future positions although it can hide the users’ velocity and direction. Commonly, the estimation of all continuous query users is not as precise as the actual positions; thus the result of cloaking may not be accurate.

Continuous query is one of the most common query types for location based services; however, current methods for protecting the privacy of continuous query are few and cost too much computation. Therefore, we study it and propose a new scheme to reduce computation of the estimation of the users’ future location in order to alleviate pressure of anonymizer servers.

3. Enhanced Greedy Algorithm Scheme

In our scheme, we adopt greedy algorithm as the basic idea and improve it by changing cloaked region form and adding adjustment algorithm; at the same time, we improved privacy and quality models to let them suit the new situation. The major notations are shown in Notations section.

3.1. Definitions. Assumption: we assume that one user only sends one query at a certain time; in other words, one query at a certain time is responding to a certain user.

Here we give some formal definitions.

Definition 1. In query $Q$, define each query from users as $Q_i$, and $Q$ is a quintuple:

$$Q = (QID, LOC, maxSpeed, T_q, T_e).$$

$QID$ is the identity of query which identifies a certain and unique query; $LOC$ is the coordinate $(x, y)$ of a place where the query is sent out; maxSpeed is the biggest moving speed of users which is represented by a vector, $V_x$ is the component of coordinate $x$, and $V_y$ is the component of coordinate $y$. $T_q$ is the time stamp to initialize a query and $T_e$ is the time when the query is expired.

Currently most of cloaking regions are represented as rectangles, marked as the coordinate of left-bottom corner and top-right corner. In our scheme, we pick cycle as cloaking region forms which is represented by center coordinate of the cycle and radius, and the definition is as follows.

Definition 2 (cloaked cycle $CS$ (cloaked cycle)). Each cloaked region represented by $CS$ is a spatial-temporal area which includes at least $k$ users:

$$CS = (CID, LOC, Ra, QS).$$

$CID$ is the identity of cloaked region which identifies a certain unique cloaked region; $LOC$ is the center coordinate of the cloaked cycle; $Ra$ is the radius of the cycle; $QS$ is a set included in the $Q$ from $CS$.

As shown in Figure 6(a), the rectangle represents the whole system, points represent the users who request query,
dashed cycles represent cloaked cycle, the black points inside
the cloaked cycle represent k anonymity users, and the gray
points outside the cycle represent the mobile users around.
Suppose mobile users U6 request 3-anonymity query; then
the dash cycle is the qualified cloaked cycle.

Definition 3 (distance cycle). We define the cycle formed by
the distance from the query users to other users at some time
as the distance cycle from query user to some certain user:

\[ R = (\text{LOC, Ra}) \tag{3} \]

LOC is the coordinate of query user S at time \( T_q \), \( R \cdot \text{LOC} = S \cdot \text{LOC} \), Ra is the distance from query user S to other user
P, suppose the query user’s coordinate is \( (x_s, y_s) \), the other
user P coordinate is \( (x_p, y_p) \), then the center coordinate of
distance cycle is \( (x_s, y_s) \), the radius is the Euclidean distance
from S to P, \( R \cdot \text{Ra} = \sqrt{(x_s - x_p)^2 + (y_s - y_p)^2} \), as shown in
Figure 6(b), and the dashed cycle is the distance from users
U6 to U9.

Definition 4 (cloaking angle (CA)). Cloaking angle is the
inclined angle between coordinate x and the beveled edge of
triangle formed by Ra’s area and radius of the cloaking cycle:

\[ \text{CA}(S, Q) = \arctan \frac{\pi \times R \cdot \text{Ra}^2}{R \cdot \text{Ra}}. \tag{4} \]

as shown in Figure 7.

From Figure 7, CA is a range from 0° to 90° (0° < CA < 90°), when S is farther from Q, and the degree of is bigger.

3.2. Cloaking Model. We improved cloaking model by pri-

vacy monitor, quality monitor, and dynamic adjuster.

Privacy monitor guarantees that, in the whole query life
cycle, any two queries in the cloaked region cannot gather to
one point.

Definition 5 (privacy monitor a). Suppose CS is a cloaked
cycle including k queries \( \{Q_1, Q_2 \cdots Q_k\} \), \( C_1, C_2 \cdots C_k \) are the
possible location area of the queries in cloaked cycle in the
whole query life cycle, and then

\[ \{ \forall t \in [T_c, \max T], \forall Q_1, Q_2 \cdots Q_k \in CS, \]

\[ C_1 \cap C_2 \cap \cdots \cap C_k \neq \phi \}. \tag{5} \]

\( T_c \) is the generation time of cloaked cycle and \( \max T = \max (Q \cdot T_c) \forall Q \in CS \). Then the cloaked cycle satisfied the
privacy monitor requirement.

Quality monitor mainly prevents the queries of the
mobile users from getting too scattered; therefore, we need to
forecast the furthest possible location according to the biggest
velocity.

Definition 6 (quality monitor β). Suppose S is a query user,
Q is an outside user, maxSpeed is Q biggest possible velocity,
and \( Q’ \) is the furthest possible position of Q in the life cycle
of S, \( |SQ’| = |SQ| + \text{maxSpeed} \times (S \cdot T_c - T_c) \); then

\[ \forall t \in [T_c, S \cdot T_c], \text{CA}(S, Q’) < \beta, \tag{6} \]

\( |SQ| = \sqrt{(x_s - x_q)^2 + (y_s - y_q)^2} \), \( S \cdot T_c \) is the expiration time
of S and \( T_c \) is present time.

3.3. Enhanced Greedy Cloaking Scheme. The main idea of
enhanced greedy cloaking algorithm (EGCA) is that when
mobile users request location based services, they send
anonymization query requests to anonymizer, the server
checks each unexpired queries from candidate cloaking set
QS one by one after receiving the query request s and judges
whether the anonymized queries (anonymized s and QS)
satisfy quality monitor model, if yes, then it inserts s into
QS, if not, it finds the next query and repeats the process
until all queries of QS are checked (Algorithm 1). Then the
server compares the numbers of QS with the users’ anonymity
requirement k; if they are bigger, then it forms a cloaked cycle
and judges whether they satisfy privacy monitor model, if yes,
then it calls the center position adapting function to adjust
the position of the cloaked cycle center, if not or the queries number is smaller than $k$, then it inserts $s$ into $QS$.

The details are as follows.

We can divide the algorithms into 4 parts: improved greedy algorithm, quality monitor model, privacy monitor model, and center adjustment algorithm (Algorithm 4).

In order to make sure that the cloaked region will not be too scattered to affect the service quality and efficiency in the query life cycle, we adopt quality monitor algorithm to avoid the above problem by predicting all possible locations of the querier according to the most biggest velocities at the initial time; thus qualified and appropriate queriers will be selected to be anonymized while unqualified ones will not be selected, as shown in Figure 8(a).

For the query shown in Figure 8(a), the arrow represents the possible biggest velocity of the queriers and the dashed cycle represents all possible positions of the queriers in the whole query life cycle. In the graph, suppose $s$ is a querier who issues 2-anonymity request; $Q_1$ and $Q_2$ are the other queriers waiting to be anonymized. $Q_1'$ and $Q_2'$ are the furthest position from $s$ to $Q_1$ and $Q_2$. For a conservative calculation, calculate the anonymization inclined angle $CA(S, Q_1')$ and $CA(S, Q_2')$ and select the queriers inside the angle region. For the example shown in the figure, $CA(S, Q_1')$ is bigger than $CA(S, Q_2')$, which means that the area to which $Q_1$ might move to is bigger, so $Q_1$ is not selected. The solid cycle is the distance cycle from the query $s$ to $Q_2$. The details of the algorithm are shown in Algorithm 2.

Algorithm 2 prevents the queriers from getting too dispersive at the initial time, while for the problem where queriers might converge to one point, we have to solve it after $k$ queriers are selected. To solve the problem, we propose another algorithm privacy monitor $\beta$ (Algorithm 3), a conservative way, and calculate all the possible future positions of each querier in the cloaked cycle and build equation set $EQ$ of possible future position of each querier to judge whether there is solution of $EQ$. If there is a solution, then the cloaked regions can be overlapped and then might
When user issues a query \( s^* \), the candidate cloaking set \( U = \{ \emptyset \} \); insert \( s \) to \( U \); for (each \( q \) in \( QS \))
{
if (\( s \cdot T_e - q \cdot T_e > \varepsilon \)) then
get the next \( q \) in \( QS \); else
if IsMeet \( Q(s, q) \) then
{
inset \( q \) to \( U \); if (|\( U \| \geq k \)) then
{
if (IsMeet \( P(U) \)) then
{Generate CS;
CenterAdj(CS);
Return CS;
}
}
}
inset \( s \) to \( QS \); }

Algorithm 1: Enhanced greedy cloaking algorithm.

/** When user issues a query \( s^* \)/
candidate cloaking set \( U = \{ \emptyset \} \);
insert \( s \) to \( U \);
for (each \( q \) in \( QS \))
{
if (\( s \cdot T_e - q \cdot T_e > \varepsilon \)) then
get the next \( q \) in \( QS \); else
if IsMeet \( Q(s, q) \) then
{
inset \( q \) to \( U \); if (|\( U \| \geq k \)) then
{
if (IsMeet \( P(U) \)) then
{Generate CS;
CenterAdj(CS);
Return CS;
}
}
}
inset \( s \) to \( QS \); }

4. Evaluation

We simulated mobile users and generated their location data by Brinkhoff Generator [15]. Input data is the city network map of OldenburgGen, output is location data of mobile users’ query which is stored in Oracle 10g database, and the algorithms are coded by Java. We mainly analyze the feasibility and effectiveness of the algorithms from three aspects: average responding time, average anonymization cost, and anonymization success rate.

4.1. Simulator Configuration (Brinkhoff Generator). In our experiment, we select Brinkhoff generator as the simulator which can simulate mobile objects for city roads; it has two basic city road maps, OldenburgGen and Sanfrancisco. We adopt OldenburgGen city network map, which includes 6105 nodes and 7035 edges, the generated data is stored in Oracle 10g database, the algorithms are coded by JAVA, and the configuration of simulation computer is CPU: Intel i3 3.20 GHz 3.20 GHz, Mem: 2 G.

Brinkhoff generator input parameters are shown in Table 1.

Brinkhoff generator is running to generate mobile users and outside objects’ location data. Figure 10 shows the 395th time stamp graph after running Brinkhoff generator. 263434 records have been generated in table moving objects, and 12078 records have been generated in table external objects. The parameters for the algorithms are configured, as shown in Table 2.

4.2. Simulation Analysis. We compare our algorithm EGCA with the original one GCA from the average responding time, average anonymity cost, and anonymity success rate.

4.3. Average Responding Time. Average responding time is the average time cost from that each query is send out until the cloaked region is received. Here it only calculates the successful anonymized queries. The time comparison of average responding time is shown in Figure 11.

From Figure II, the average responding time varies with the degree \( k \) of anonymity. If \( k \) is bigger, the average responding time is longer and rise of freight rate, in that with \( k \) becomes bigger, more mobile queries need to be solved. It needs more computation to get the distance cycles, thus it alleviates the anonymizer and responding time is longer.
/* judge whether the querier \( q \) satisfies quality monitor requirement */
Function (Query \( s, q \))
    const PI = 3.1415926
    double distance = 0;
    distance = the length from \( s \) to the center of cycle \( q \)
    distance += the length from \( s \) to the center of cycle \( q' \)
    If \( CA(s, q) < CA(s, q') \) then
        return true
    else
        return false;
}

Algorithm 2: Quality monitor \( \alpha \).

Function (CS cs)
{ Define \( EQ[cs\cdot QS\cdot count] \);
    double \( x, y \);
    \( i = 1 \);
    for (each \( q \) in \( cs\cdot QS \))
        push \((x - x_q)^2 + (y - y_q)^2 = (maxSpeed \cdot (max T - T_s))^2\) in \( EQ[i++] \);
    for (equation Group in \( EQ \))
        if (have solution of equation Group) then
            return false
        else
            return true;
}

Algorithm 3: Privacy monitor \( \beta \).

Function (CS cs, Query \( s \))
{
    For (each \( q \) in \( cs\cdot QS \))
    {
        if \( x_q < x_{\min} \) then \( x_{\min} = x_q \);
        if \( x_q > x_{\max} \) then \( x_{\max} = x_q \);
        if \( y_q < y_{\min} \) then \( y_{\min} = y_q \);
        if \( y_q > y_{\max} \) then \( y_{\max} = y_q \);
        \( x = x_{\max} + x_{\min} - 2x_s \);
        \( y = y_{\max} + y_{\min} - 2y_s \);
        \( x = \text{Random}(0, x) \);
        \( y = \text{Random}(0, y) \);
        if \( x > 0 \) then
            \( cs\cdot x += x \) else \( cs\cdot x -= x \);
        if \( y > 0 \) then \( cs\cdot y += y \) else \( cs\cdot y -= y \);
        Return cs;
    }
}

Algorithm 4: Center adjustment algorithm.

In the figure, the average responding time of ECGA is less than CGA, in which ECGA predicts all possible positions of the mobile queries which satisfy quality monitor and privacy monitor requirement at the initial time. Thus in the whole query life cycle, each query moving position is in a certain area. When anonymizer generates cloaked cycle, it only needs to update the current position of each query. While CGA needs to check the border of rectangle cloaked region and calculate the space anonymizing degree of each query by definite integration, besides, it needs to check the length and width of rectangle for quality model and solve the set of
twocurves;however,ECGAonlyneeds to check whether the regions of all the possible positions of the query inside the cloaked region have intersection that only needs to solve one equation set. Therefore, the average responding time of ECGA is less than CGA.

4.4. Average Anonymization Costs. Average anonymization costs represent the costs that the anonymizer solves the query requests after the requests are sent out. In CGA, the average anonymization costs are defined as the average perimeters of the cloaked rectangle in the query life cycle, when the average perimeter is bigger, then the cost is higher. While the average anonymization costs in ECGA is defined as the average perimeter of cloaked cycle in the query life cycle, Figure 12 shows the comparison.

For Figure 12, the average anonymity cost of both algorithms rises with the increasing of \(k\). In the figure, we can find that two curves are intersectant; before the intersection point, average anonymization costs of ECGA is higher than CGA, while after the intersection point, ECGA is lower in average anonymization costs. In both algorithms we use average perimeter to calculate the average anonymization costs, while both perimeters and diameters are decided by two furthest queries along coordinates \(x\) and \(y\), which means that when the region perimeter is \(6.28R\) (\(R\) is half of the distance between two furthest queries), both perimeters are identical. When the region perimeter is smaller than \(6.28R\), the average anonymization costs of ECGA are higher, and when the perimeter is bigger than \(6.28R\), costs are lower than CGA.

4.5. Anonymization Success Rate. Anonymization success rate means the ratio of the number of queries which return cloaked cycle with the numbers of all request queries. It reflects the feasibility of algorithm. Figure 13 gives a comparison columnar graph of the anonymization success rate of ECGA and GCA and the anonymity degree \(k\) is from 3 to 8.

As shown in Figure 13, the success rate of ECGA anonymization is equal or a little higher than GCA when \(k\) is relatively small. When \(k\) becomes bigger, the success rate of ECGA anonymization is a littel lower than GCA. When EGCA searches the qualified mobile queries, the quality
monitor is defined to be relative fixed in strict way while GCA is changing. When \( k \) is small, the qualified queries are relatively few; when \( k \) is big, the fixed condition limits some mobile queries, so the success rate starts to decline and becomes lower than GCA. However, from the simulation, even \( k \) is 10, the success rate is still above 90%, and thus it is acceptable for practice in real application.

5. Conclusion

In this paper, we analyze the existing location privacy protection systems and algorithms for location-based services, considering their disadvantages of slow responding time and high anonymization costs; we propose a new enhanced greedy cloaking algorithm which predicts a cloaking area at the initial query to be the cloaking region in the whole query lifetime by comprehensive control of privacy monitor, quality monitor, and dynamic adjuster. Privacy monitor and quality monitor charge the privacy protection level and service quality degree, respectively; dynamic adjuster can adjust the cycle center point dynamically. We employ cycle as cloaking region form which it can effectively alleviate the computation overhead. And we compare it with the earlier algorithm on three aspects. The experiment result shows that the enhanced greedy cloaking algorithm is better than the original greedy algorithm on average responding time or anonymization cost.

Notations

- \( Q \): Query from mobile user
- \( QID \): Id of a query
- \( LOC \): Current location when a query is issued
- \( maxSpeed \): Biggest moving speed of users which is represented by a vector
- \( T_a \): The time stamp to initialize a query
- \( T_c \): The time when the query is expired
- \( CS \): Cloaking cycle: a spatial temporal area which includes at least \( k \) users
- \( CID \): The identity of cloaked region which identify a certain unique cloaked region
- \( Ra \): The radius of the cloaking cycle
- \( QS \): A set included in the \( Q \) from \( CS \)
- \( CA \): The inclined angle between coordinate \( x \) and the beveled edge of triangle formed by \( Ra \)’s area and radius of the cloaking cycle
- \( \alpha \): Quality monitor
- \( \beta \): Privacy monitor.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (no. 6110092) and Research Fund for the Doctoral Program of Higher Education of China (no. 20112302120074) and was partially supported by Shenzhen Strategic Emerging Industry Development Foundation (nos. JCYJ2012063151032592 and ZDSY2012063125016389), National Key Technology R&D Program of MOST China under Grant no. 2012BAK17B08, and the National Common-wealth Technology R&D Program of AQSIO China under Grant no. 201310087. The authors thank the reviewers for their comments.

References

[1] Y. Zheng, Y. Chen, X. Xie, and W.-Y. Ma, “GeoLife2.0: a location-based social networking service,” in Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM ’09), pp. 357–358, May 2009.
[2] Y. Zheng, “Tutorial on location-based social networks,” in Proceedings of the 21st International Conference on World Wide Web (WWW ’12), p. 12, 2012.
[3] P. Li, H. Hu, and J. Xu, “Nearby friend alert: location anonymity in mobile geosocial networks,” IEEE Pervasive Computing, vol. 12, no. 4, pp. 62–70, 2013.
[4] B. Gedik and L. Liu, “Protecting location privacy with personalized k-anonymity: architecture and algorithms,” IEEE Transactions on Mobile Computing, vol. 7, no. 1, pp. 1–18, 2008.
[5] A. Khoshgozaran, C. Shahabi, and H. Shirani-Mehr, “Location privacy: going beyond K-anonymity, cloaking and anonymizers,” Knowledge and Information Systems, vol. 26, no. 3, pp. 435–465, 2011.
[6] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux, “Unraveling an old cloak: K-anonymity for location privacy,” in Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society (WPES ’10), pp. 115–118, ACM, October 2010.
[7] J. Jamsa and M. Luimula, “Advanced car navigation: future vehicle instrumentation for situation-aware services,” in Proceedings of the 12th IEEE International Conference on Mobile Data Management (MDM ’11), pp. 7–10, June 2011.
[8] D. Ghosh, A. Joshi, T. Finin et al., “Privacy control in smart phones using semantically rich reasoning and context modeling,” in Proceedings of the IEEE Symposium on Security and Privacy Workshops (SPW ’12), pp. 82–85, 2012.
[9] Y. Wang, L. Wang, and B. C. M. Fung, “Preserving privacy for location-based services with continuous queries,” in Proceedings of the IEEE International Conference on Communications (ICC ’09), pp. 1–5, June 2009.
[10] A. R. Beresford and F. Stajano, “Location privacy in pervasive computing,” IEEE Pervasive Computing, vol. 2, no. 1, pp. 46–55, 2003.
[11] C. Y. Chow and M. F. Mokbel, “Enabling private continuous queries for revealed user locations,” in Advances in Spatial and Temporal Databases, pp. 258–275, Springer, Berlin, Germany, 2007.
[12] C.-Y. Chow, M. F. Mokbel, J. Bao, and X. Liu, “Query-aware location anonymization for road networks,” GeoInformatica, vol. 15, no. 3, pp. 571–607, 2011.
[13] X. Pan, X. Hao, and X. Meng, “Privacy preserving towards continuous query in location-based services,” Computer Research and Development, vol. 47, no. 1, pp. 121–129, 2010.
[14] Y. Wang, L. Wang, and B. C. M. Fung, “Preserving privacy for location-based services with continuous queries,” in Proceedings
of the IEEE International Conference on Communications (ICC ’09), pp. 1–5, June 2009.

[15] J. Muckell, P. W. Olsen, J. H. Hwang et al., "A framework for efficient and convenient evaluation of trajectory compression algorithms," in Proceedings of the 4th International Conference on Computing for Geospatial Research and Application, pp. 24–31, IEEE, 2013.
Submit your manuscripts at http://www.hindawi.com