Geoinformatica (2011) 15:351–380
DOI 10.1007/s10707-009-0099-y

Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments

Chi-Yin Chow · Mohamed F. Mokbel · Xuan Liu

Received: 27 January 2009 / Revised: 23 September 2009 /
Accepted: 22 October 2009 / Published online: 14 November 2009

Abstract This paper tackles a privacy breach in current location-based services (LBS) where mobile users have to report their exact location information to an LBS provider in order to obtain their desired services. For example, a user who wants to issue a query asking about her nearest gas station has to report her exact location to an LBS provider. However, many recent research efforts have indicated that revealing private location information to potentially untrusted LBS providers may lead to major privacy breaches. To preserve user location privacy, spatial cloaking is the most commonly used privacy-enhancing technique in LBS. The basic idea of the spatial cloaking technique is to blur a user's exact location into a cloaked area that satisfies the user specified privacy requirements. Unfortunately, existing spatial cloaking algorithms designed for LBS rely on fixed communication infrastructure, e.g., base stations, and centralized/distributed servers. Thus, these algorithms cannot be applied to a mobile peer-to-peer (P2P) environment where mobile users can only communicate with other peers through P2P multi-hop routing without any support of fixed communication infrastructure or servers. In this paper, we propose a spatial cloaking algorithm for mobile P2P environments. As mobile P2P environments have many unique limitations, e.g., user mobility, limited transmission range, multi-hop communication, scarce communication resources, and network partitions, we propose three key features to enhance our algorithm: (1) An *information sharing*

This work is supported in part by the National Science Foundation under Grant Numbers IIS-0811998, IIS-0811935, and CNS-0708604, and by Microsoft Research Gift.

C.-Y. Chow · M. F. Mokbel (✉)
Department of Computer Science and Engineering,
University of Minnesota - Twin Cities, 200 Union Street SE, Minneapolis, MN 55455, USA
e-mail: mokbel@cs.umn.edu

C.-Y. Chow
e-mail: cchow@cs.umn.edu

X. Liu
IBM Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532, USA
e-mail: xuanliu@us.ibm.com

scheme enables mobile users to share their gathered peer location information to reduce communication overhead; (2) A *historical location scheme* allows mobile users to utilize stale peer location information to overcome the network partition problem; and (3) A *cloaked area adjustment scheme* guarantees that our spatial cloaking algorithm is free from a "center-of-cloaked-area" privacy attack. Experimental results show that our P2P spatial cloaking algorithm is scalable while guaranteeing the user's location privacy protection.

Keywords Spatial cloaking · Location anonymization · Location-based services · Mobile peer-to-peer computing

1 Introduction

Location-based services (LBS) can provide a wide variety of important services for mobile users that have been proven through many commercial products or research prototypes. Examples of these services include *transportation services* (e.g., "What is the shortest route from my current location to my home"), *convenience services* (e.g., "Where is my nearest grocery store"), and *emergency control* (e.g., "Dispatch the nearest ambulance to the patient"). Since LBS is provided for users based on their exact location information, a major threat about the user's location privacy has been raised. Recently, spatial cloaking has been widely used to tackle such a privacy breach in LBS. The basic idea of the spatial cloaking technique is to blur a user's exact location into a cloaked area such that the cloaked area satisfies the user specified privacy requirements (e.g., [1, 4, 7, 11, 15, 19, 23, 24]). The most popular privacy requirements for the spatial cloaking technique are K-anonymity, i.e., a cloaked area contains at least K users, and minimum area A_{min}, i.e., the size of a cloaked area is at least A_{min}. Since a location-based database server does not know the user's exact location information, the database server can only return an answer set that includes the exact answer to the user (e.g., [14, 15, 19]).

A mobile peer-to-peer (P2P) network is a highly ad-hoc environment in which mobile users can only communicate with other peers through multi-hop routing without any support of fixed communication infrastructure or centralized/distributed servers. There are many unique limitations in the mobile P2P environment, e.g., user mobility, limited transmission range, multi-hop communication, scarce communication resources, and network partitions.[1] In terms of system architecture, existing spatial cloaking techniques can be classified into three main categories, *centralized* (e.g., [1, 7, 11, 15, 19, 23, 24]), *distributed* (e.g., [9, 10]), and *peer-to-peer* approaches (e.g., [4]). Since the spatial cloaking algorithms proposed for the *centralized* or *distributed* approach rely on fixed communication infrastructure and centralized/distributed servers, they cannot be applied to the mobile P2P environment. To our best knowledge, our previous work is the only spatial cloaking algorithm for the mobile P2P environment [4]. The main idea of the P2P spatial cloaking algorithm is that when a mobile user wants to obtain services from an LBS provider, she collaborates with other peers via multi-hop communication to blur her

[1] In a partitioned network, mobile users are partitioned into disjoint networks, in which a mobile user is only able to communicate with other peers residing in her network partition.

location into a cloaked area. Our algorithm guarantees that the cloaked area satisfies the user's K-anonymity and minimum area A_{min} privacy requirements. Then, the user sends her location-based query along with the cloaked area to the LBS provider to obtain her desired services. Since the location-based database server does not know the exact user location, it can only return an answer set that includes the exact answer to the user. Thus, after the user gets the answer set from the database server, she has to compute the exact answer from the answer set.

In this paper, we propose three key features to enhance the scalability, efficacy and privacy protection of our P2P spatial cloaking algorithm. Since the mobile P2P environment has limited communication resources and constrained transmission range, excessively searching the network for peers would pose a scalability issue. To this end, we propose an *information sharing scheme* for our P2P spatial cloaking algorithm to enable mobile users to share their gathered peer location information with nearby peers. If the mobile user can get enough peer location information from a peer, she does not need to search the network; and therefore, the *information sharing scheme* can reduce communication overhead. In addition, the mobile user may encounter a network partition problem in the mobile P2P environment, i.e., the number of users residing in her network partition is less than her required anonymity level, i.e., K, she cannot find enough peer location information to satisfy her K-anonymity privacy requirement. To alleviate the network partition problem, we design a *historical location scheme* that allows users to utilize the peer location information cached by the peers residing in their network partition. We use a consecutive approach to adjust such stale location information to capture its uncertainty. Furthermore, the mobile user needs to search the network for an adequate number of peers to satisfy her required anonymity level. Since the peer search process usually starts at the user and spreads out from her nearby peers to farther peers, the user may be close to the center of the cloaked area. Thus, an adversary could guess that the user who is the closest to the center of a cloaked area is the actual query issuer, i.e., a "center-of-cloaked-area" privacy attack. To avoid such a privacy breach, we propose a *cloaked area adjustment scheme* that adjusts a cloaked area such that the probability of the actual query issuer being the closest to the center of a cloaked area is $1/K$.

We evaluate the performance of our proposed features through simulated experiments. The experimental results show that these features enhance the scalability of our P2P spatial cloaking algorithm while guaranteeing the user's location privacy protection. In general, the contributions of this paper can be summarized as follows:

- We propose a spatial cloaking algorithm for mobile peer-to-peer environments. (Section 4.1)
- We introduce an *information sharing scheme* for our spatial cloaking algorithm to enable mobile users to share their gathered peer location information with other peers in order to reduce communication overhead. (Section 4.2)
- We design a *historical location scheme* for our algorithm to overcome the network partition problem. (Section 4.3)
- We propose a *cloaked area adjustment scheme* for our algorithm to avoid the "center-of-cloaked-area" privacy attack. (Section 4.4)
- We experimentally evaluate our P2P spatial cloaking algorithm with the three enhancement schemes. The experimental results show that the enhanced algorithm is scalable while guaranteeing the user's location privacy protection. (Section 6)

The rest of this paper is organized as follows. Section 2 highlights related works. Section 3 gives our system model. Section 4 describes our peer-to-peer spatial cloaking algorithm and the three enhancement features. Section 5 presents the privacy-aware query processing for anonymous location-based services. Experimental results are depicted in Section 6. Section 7 concludes this paper.

2 Related works

Privacy-preserving techniques for location privacy have been widely studied. These techniques are based on one of the following concepts. (a) *False locations*. Users protect their location privacy by reporting either fake locations [26] or their exact locations with a set of fake locations, termed *dummies* [17], to a location-based database server. (b) *Space transformation*. The user location information and data are transformed into another space in which their exact [8, 25] or approximate [16] spatial relationships are maintained to answer location-based queries. (c) *Spatial cloaking*. The main idea of the spatial cloaking technique is to blur a user's exact location into a cloaked area that satisfies the user's privacy requirements [1, 3, 4, 6, 7, 9–11, 13, 15, 19, 23, 24], e.g., K-anonymity [21] (i.e., the cloaked area contains at least K users) and minimum area A_{min} (i.e., the cloaked area size is at least A_{min}). Among these concepts, we employ the spatial cloaking technique to protect the user location privacy because this technique is the most popular one and it supports many environment settings, e.g., centralized [1, 7, 11, 15, 19, 23, 24], distributed [9, 10], peer-to-peer [4], and wireless sensor networks [12], and many problem settings, e.g., snapshot queries [1, 4, 7, 9–11, 15, 19], continuous queries [3, 23], and trajectories [24].

In terms of architecture models, existing spatial cloaking techniques can be categorized into three models, *centralized*, *distributed* and *peer-to-peer*. For the *centralized* architecture model [1, 7, 11, 15, 19, 23, 24], a trusted third party, termed *location anonymizer*, is placed between the user and the location-based service provider. The *location anonymizer* is responsible for blurring users' exact locations into cloaked areas that satisfy their privacy requirements, and for communicating with the service provider. This architecture model could pose a scalability issue because it requires all the mobile users to periodically report their exact locations to the *location anonymizer*. Also, storing the user's exact location at a server could pose a privacy breach, i.e., a single point of attacks [9, 10, 27]. For the *distributed* architecture model [9, 10], the users maintain a complex data structure to anonymize their location information through fixed communication infrastructure, i.e., base stations. However, such a complex data structure leads to difficulties to apply this model to highly dynamic location-based mobile applications [27]. For the *peer-to-peer* model, to our best knowledge, our previous work is the only spatial cloaking algorithm for this architecture model [4]. The basic idea is that mobile users are able to work together to blur their locations into cloaked areas without using any fixed communication infrastructure or centralized/distributed servers.

Since the user's location information is blurred into cloaked areas, a query processor embedded inside a location-based database server must have the ability to deal with location-based queries with cloaked areas. The state-of-the-art privacy-aware query processor can deal with nearest-neighbor queries with either rectangular cloaked areas [5, 14, 19] or circular cloaked areas [15]. Since the query processor does

not know the exact location of the query, it can only determine an answer set that includes the exact answer to the query [14, 15, 19] or an approximate answer set with a certain quality guarantee [5]. Among the algorithms returning the exact answer within the answer set to the user, the works [14, 15] compute the minimal answer set while the work [19] computes a superset of the answer set. Then, the database server sends the answer set to the user, and the user computes the exact answer from the answer set (or the best answer from the approximate answer set). In this paper, we adopt the work [14] as our privacy-aware query processor because it minimizes the communication overhead of sending the minimal answer set from the database server to the user while guaranteeing the user can get the exact answer.

3 System model

Figure 1 depicts the system architecture of our peer-to-peer (P2P) spatial cloaking algorithm that consists of two entities, *mobile users* and *location-based database servers*. We will first discuss our privacy threat model and privacy settings in *user privacy profiles*, and then describe each entity in our system.

Privacy threat model We assume that mobile users are trusted, so they do not use their gathered peer location information to attack our system. However, we do not have any assumption about the trustworthiness of the location-based service providers. Thus, we assume that an adversary can utilize the information gathered by the service provider to make privacy attacks. In addition, we only focus on snapshot location-based queries, and each query is assigned a unique pseudonymous identity that is completely unrelated to the user's personal identity in order to ensure the pseudonymity of the user's location information [20].

User privacy profiles Each user specifies her privacy requirements in a *privacy profile* in a form of (K, A_{min}), where K indicates the required anonymity level and A_{min} indicates the required minimum area of her cloaked areas. In other words, the user wants to find a cloaked area that includes at least K users and has an area of

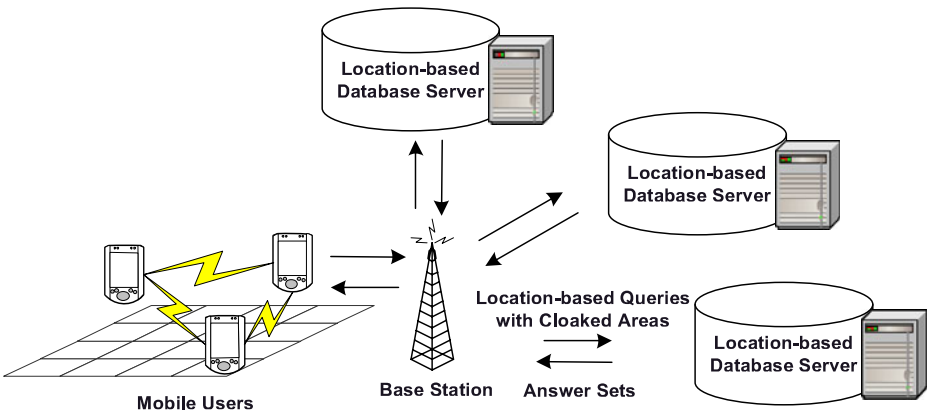

Fig. 1 System architecture

at least A_{min}. A_{min} is particularly useful in a dense area where a large K would not achieve high privacy protection. For example, a user in a stadium with $K = 100$ may result in a very small cloaked area. It is important to note that the user can change her privacy profile at any time to guarantee that her specified privacy settings achieve her desired privacy protection in different situations.

The privacy profile can be extended to support temporal, spatial, and/or interdependent constraints on the required anonymity level and the required cloaked area size.

- *Temporal constraint.* The user can specify her privacy requirements for different time intervals. For example, a user is willing to reveal her location information during office hours, e.g., ($K = 1$, $A_{min} = 0$ mile) between 8:00 AM and 5:00 PM, but she needs high privacy protection after office hours, e.g., ($K = 100$, $A_{min} = 5$ miles) after 5:00 PM.
- *Spatial constraint.* The user can specify her privacy requirements for different geographic regions. For example, a user needs a small A_{min} in downtown areas, e.g., ($K = 50$, $A_{min} = 1$ mile), but a large A_{min} in rural areas, e.g., ($K = 50$, $A_{min} = 10$ miles). This is because a high density of geographic features in a downtown area can make the adversary very difficult to infer anything about the user even a small cloaked area of a couple of city blocks.
- *Interdependent constraint.* The user can specify an interdependence between the anonymity level and the size of a cloaked area. For example, a user can specify a maximum area A_{max} requirement of her cloaked areas, i.e., she is happy with a cloaked area of a size A_{max} regardless of the number of users within the cloaked area. A_{max} also ensures the query utility of our spatial cloaking algorithm.

Since it is straightforward to extend our P2P spatial cloaking algorithm to support temporal, spatial, and/or interdependent constraints, we only discuss how our algorithm finds cloaked areas satisfying both K and A_{min} in this paper.

Mobile users Each mobile user is equipped with two wireless network interface cards; one of them is dedicated to connect to a mobile base station to communicate with *location-based database servers*, while the other one is devoted to communicate with other peers via multi-hop routing without any support of fixed communication infrastructure or centralized/distributed servers. This multi-interface approach has been adopted in location-based services (e.g., [4, 18]) and mobile peer-to-peer information access applications (e.g., [2, 22]). Each user is also equipped with a positioning device, e.g., GPS, to determine her location that is represented as a coordinate (x, y). It is important to note that we do not have any assumption about the transmission range of the user mobile device, i.e., the mobile users can have different transmission ranges, and the network topology.

Location-based database servers A privacy-aware query processor embedded inside the *location-based database server* has the ability to deal with location-based queries with cloaked areas. When a mobile user wants to obtain services from a location-based service provider, she executes our P2P spatial cloaking algorithm to blur her location into a cloaked area that satisfies her privacy requirements. Then, the user sends her location-based query along with the cloaked area to the database server. Since the query processor does not know the exact user location, it can only

compute an answer set that includes the exact answer to the user. Then, the database server sends the answer set to the user, and the user computes the exact answer from the answer set.

4 Peer-to-peer spatial cloaking algorithm

We now present our spatial cloaking algorithm in mobile peer-to-peer (P2P) environments where no fixed communication infrastructure or centralize/distributed servers are available. Thus, the mobile users are only able to collaborate with each other through multi-hop routing to execute our algorithm. Such a highly ad-hoc mobile network poses many limitations to the computing environment, e.g., user mobility, limited transmission range, multi-hop communication, scarce communication resources, and network partition problems. In this section, we first describe our P2P spatial cloaking algorithm (Section 4.1). Then, we present the three key features proposed for our algorithm. Section 4.2 describes the *information sharing scheme* that enables users to share their gathered peer location information with nearby peers in order to reduce communication overhead. Section 4.3 gives the *historical location scheme* that allows users to utilize stale peer location information to alleviate the network partition problem. Section 4.4 presents the *cloaked area adjustment scheme* that guarantees our algorithm to be free from the "center-of-cloaked-area" privacy attack. We start by assuming that the network partition problem does not take place, i.e., all mobile users can communicate with each other through multi-hop routing in the network. The network partition problem will be addressed in Section 4.3.

4.1 Spatial cloaking algorithm

The basic idea of our P2P spatial cloaking algorithm is that a mobile user communicates with other peers via multi-hop routing to find at least $K-1$ peers. Then, the user determines a cloaked area that includes the $K-1$ nearest peers and herself. The cloaked area is K-anonymous because the user is indistinguishable among K users within the cloaked area. After satisfying the K-anonymity privacy requirement, the user extends the cloaked area to have an area of at least A_{min}, in order to satisfy the minimum area privacy requirement. To obtain location-based services, the user sends her location-based query along with the result cloaked area as her blurred location information to a database server. We will discuss how the database server computes an answer set that includes the exact answer to the user based on the cloaked area in Section 5.

Algorithm Algorithm 1 depicts the pseudo code of our P2P spatial cloaking algorithm. Figure 2 depicts a running example to illustrate the algorithm where 15 mobile users are labeled from m_1 to m_{15}. m_8 who executes the algorithm is represented by a triangle, and other peers are represented by circles. We assume that m_8's required anonymity level is five, i.e., $K=5$, and her required minimum area is A_{min}. In general, the P2P spatial cloaking algorithm consists of two main steps.

Step 1: ***Peer search step*** The user U starts this step by enlisting her neighbor peers for help. First, U broadcasts a request to her neighbor peers, i.e., the hop

Algorithm 1 Peer-to-Peer Spatial Cloaking

1: **function** P2PSpatialCloaking(User U)
2: //***Step 1: Peer Search Step***
3: $List \leftarrow \{\emptyset\}$
4: $h \leftarrow 1$
5: **while** $NumUser(List) < U.K - 1$ **do**
6: Broadcast a request to the peers within h hop(s) from U
7: $List \leftarrow List \cup$ {the received peer location information}
8: $h \leftarrow h + 1$
9: **end while**
10: //***Step 2: Cloaked Area Step***
11: $S \leftarrow \{U\} \cup$ {the $K - 1$ nearest peers of U in $List$}
12: $A \leftarrow$ a minimum bounding rectangle of all users in S
13: **if** $Area(A) < U.A_{min}$ **then**
14: $\alpha \leftarrow \frac{-2(w+l)+\sqrt{4(w+l)^2-16(Area(A)-U.A_{min})}}{8}$, where w and l are the width and length of A, respectively
15: Expand each edge of A by α
16: **end if**
17: **return** A

distance between U and her neighbor peers is one ($h = 1$). Each neighbor peer replies her identity and location information to U. Then, U stores the received peer information in a list $List$. If U has at least $K - 1$ neighbor peers, U gets enough peer information, so U proceeds to the next step. However, if U does not have at least $K - 1$ neighbor peers, U has to enlist multi-hop peers for help. U increases h by one, i.e., $h = 2$, and broadcasts the request to the peers within two hop distance. When a peer receiving the request, if $h > 1$, the peer replies her identity and location information to U, decreases h by one, and forwards the request with the updated h to her neighbor peers. However, if $h = 1$, the peer simply replies her identity and location information to U. U also stores the received peer information in $List$. In case that U still cannot find an adequate number of peers within h hop distance, i.e., $NumUser(List) < K - 1$, where $NumUser(List)$ returns the total number of peers included in $List$, U repeats this peer search process until U finds at least $K - 1$ peers, as depicted in Lines 5 to 9 in Algorithm 1. After U finds enough peer location information, it proceeds to the *cloaked area step*.

Figure 2a, b illustrate the *peer search step*, where m_8 executes the algorithm. The transmission range of m_8 is represented by a dotted circle (Fig. 2a). Thus, m_8 finds three neighbors peers, m_5, m_7, and m_9, that are represented by black circles, when m_8 searches for her neighbor peers (i.e., $h = 1$). Since m_8 cannot find an adequate number of peers to satisfy her required anonymity level, i.e., $m_8.K = 5$, m_8 has to enlist more peers for help. m_8 broadcasts a request with $h = 2$ to her neighbor peers. After the neighbor peers m_5, m_7, and m_9 receive the request with $h > 1$, they send their identity and location information to m_8, decrease h by one, and forward the request with $h = 1$ to their neighbor peers. In Fig. 2b, the transmission ranges of m_5,

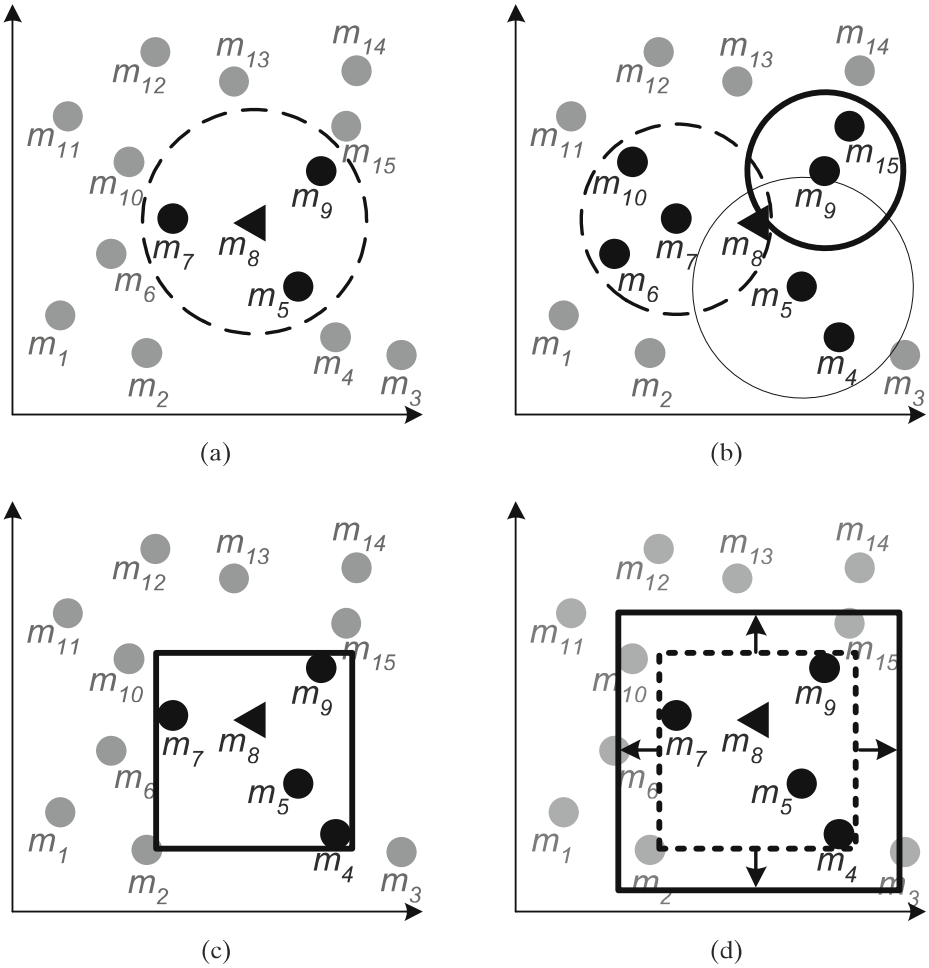

Fig. 2 Example of the peer-to-peer spatial cloaking algorithm. **a** One-hop peer search. **b** Two-hop peer search. **c** Peer selection. **d** Area expansion

m_7, and m_9 are represented by thin, dotted, and bold circles, respectively. Thus, the two-hop peers m_4, m_6, m_{10}, and m_{15} receive the request with $h = 1$. Since $h = 1$, they simply send their identity and location information to m_8. After the two-hop search, m_8 has the location information of seven peers, m_4, m_5, m_6, m_7, m_9, m_{10}, and m_{15}, m_8 has enough peer location information to proceed to the *cloaked area step*.

Step 2: ***Cloaked area step*** This step takes the peer location information stored in *List* from the previous step as an input, and determines a cloaked area A that satisfies both the user K-anonymity and minimum area privacy requirements, i.e., $NumUser(A) \geq K$ and $Area(A) \geq A_{min}$, where $Area(A)$ returns the area of A. We find a set of users S that includes U and the $K-1$ nearest peers to U in *List* (Line 11 in Algorithm 1). Then, A is a minimum bounding rectangle of U and the selected peers in S (Line 12 in

Algorithm 1). A is represented by its bottom-left vertex (x_s, y_s) and top-right vertex (x_e, y_e).

Although A already satisfies the K-anonymity privacy requirement, we still need to check for the minimum area privacy requirement. If the area of A is larger than or equal to A_{min}, i.e., $Area(A) \geq A_{min}$, the algorithm simply returns A as U's blurred location information. However, if $Area(A) < A_{min}$, we extend each edge of A by a distance α such that the area of the extended A is equal to A_{min}. Let w and l be the width (i.e., $x_e - x_s$) and height (i.e., $y_e - y_s$) of A, respectively. We can determine α by solving the following equation:

$$(w + 2\alpha)(l + 2\alpha) = A_{min}$$

$$4\alpha^2 + 2(w + l)\alpha + w \times l - A_{min} = 0$$

$$4\alpha^2 + 2(w + l)\alpha + Area(A) - A_{min} = 0 \quad (1)$$

Since $Area(A) - A_{min} < 0$, $[2(w + l)]^2 - 4(4)(Area(A) - A_{min}) \geq 0$, i.e., the discriminant of Eq. 1 is non-negative; hence, α is equal to the non-negative root of Eq. 1, i.e., $\alpha = \frac{-2(w+l)+\sqrt{4(w+l)^2-16(Area(A)-A_{min})}}{8}$, as depicted in Line 14 in Algorithm 1. After determining α, we extend each edge of A by α to form a new cloaked area that satisfies both the K-anonymity and minimum area privacy requirements. Thus, the bottom-left and top-right vertices of the extended A are $(x_s - \alpha, y_s - \alpha)$ and $(x_e + \alpha, y_e + \alpha)$, respectively.

Figure 2c, d illustrate the *cloaked area step*, where m_8 knows the location information of seven peers, i.e., $List = \{m_4, m_5, m_6, m_7, m_9, m_{10}, m_{15}\}$. We find a set of users S that includes m_8 and the four nearest peers of m_8 in $List$, i.e., m_4, m_5, m_7, and m_9. The selected peers in S are represented by black circles in Fig. 2c. Then, we determine a minimum bounding rectangle of the users in S as a cloaked area A that is represented by a rectangle. In this example, we assume that the area of A is less than A_{min}. Thus, we determine α and extend each edge of A by α. In Fig. 2d, the distance of α is indicated by arrows, and the extended A is represented by a rectangle, while the original A is represented by a dotted rectangle.

4.2 Information sharing scheme

As described in the previous section, when a mobile user wants to obtain anonymous location-based services from a service provider, she executes our P2P spatial cloaking algorithm to blur her location into a cloaked area. Due to scarce communication resources in mobile P2P environments, excessively searching the network for peers could pose a scalability issue. For example, if many nearby users search the network for peers within a short time period, they would suffer from long searching time. In the mobile P2P network, the user has to enlist her neighbor peers for help to forward messages to multi-hop peers, in order to search the network for enough peer information. Thus, a set of nearby mobile users would have a similar set of peers within the same hop distance. To this end, we propose the *information sharing scheme* for our P2P spatial cloaking algorithm to improve system scalability. The main idea is to enable a group of nearby mobile users to share their gathered peer location information with others. If the user obtains enough peer location information from her neighbor peer, she can blur her location without performing

Algorithm 2 Information Sharing Scheme

```
1: function InformationSharingScheme(User U, Tolerance tol_s)
2: //Step 1: Peer Search Step
3: List stores the received peer location information of the last peer search
4: if the timestamp of the last peer search < t_now − tol_s then
5:    Request the neighbor peers to turn in their List size and the timestamp of their
      last peer search t_search
6:    if some neighbor peer has List with a size of at least U.K and the timestamp
      of her last peer search ≥ t_now − tol_s then
7:       Select the neighbor peer P with the latest peer search timestamp
8:       Request P to turn in the cached location information of the K-nearest peers
         to U
9:       List ← the received peer location information
10:   else
11:      return P2PSpatialCloaking(User U) (i.e., Algorithm 1)
12:   end if
13: end if
14: //Step 2: Cloaked Area Step
15: Adjust the peer location information in List
16: A ← a minimum bounding rectangle includes U and the adjusted location region
    of the K − 1 nearest P of U in List
17: Execute Lines 13 to 16 in Algorithm 1 to ensure that A satisfies the minimum
    area privacy requirement
18: return A
```

the *peer search step* in Algorithm 1. Thus, the *information sharing scheme* can reduce communication and computational overhead.

Algorithm Algorithm 2 depicts the pseudo code of the *information sharing scheme*. In this scheme, after the mobile user has found enough peer information through the *peer search step* in Algorithm 1, she maintains the received peer information in *List* and records the timestamp, t_{search}, when the last *peer search step* started. Since this scheme only provides the cached peer location information for the user, the information could be cached a long time ago. The older the cached information, the lower the quality of a cloaked area (i.e., the cloaked area size) will be obtained. To this end, the *information sharing scheme* enables the user to control the staleness of the cached information through a user specified parameter tol_s. In other words, the user only utilizes the peer's stored information cached not earlier than tol_s. The *information sharing scheme* can be incorporated into our P2P spatial cloaking algorithm (i.e., Algorithm 1) with the following modifications.

Step 1: ***Peer search step*** The user U starts this step by checking whether she can use her cached information and/or her neighbor peers' cached information to blur her location. U first checks the freshness of her cached peer information. If the timestamp of the last peer search is larger than or equal to $t_{now} - tol_s$, where t_{now} is the current time, U can reuse her cached peer information. Otherwise, U requests her neighbor peers to reply with the

size of their *List* along with t_{search} (Line 5 in Algorithm 2). If more than one neighbor peer caches enough peer location information and $t_{search} \geq t_{now} - tol_s$, U selects the peer P caching the freshest information, i.e., the largest t_{search} (Line 7 in Algorithm 2). Then, U requests P to turn in the $K-1$ nearest peer information to U. However, if no neighbor peer caches enough peer location information or the cached information is too stale, U simply executes the original *peer search step* in Algorithm 1 (Line 11 in Algorithm 2). After U receives enough peer information, U stores the peer information in *List*.

Step 2: ***Cloaked area step*** Due to user mobility, i.e., mobile users are continuously moving, the only modification to this step is to capture the location uncertainty of the cached peer information. The basic idea is that given a peer's location cached at time t, we use a conservative approach to determine a location region that includes all possible locations of the peer at current time t_{now}. Since the current location of the peer could be at any point within a circular area centered at the peer's cached location with a radius of $(t_{now} - t) \times v_{max}$, where v_{max} is the maximum possible speed of the peer and t is the time when the peer's location information was cached, such a circular area constitutes the peer's *adjusted location region*. For a peer P with an adjusted location region, we consider the maximum distance between U and P's adjusted location region, i.e., $d_{max}(U, P) = d(U, P) + (t_{now} - t) \times v_{max}$, as their distance. In reality, v_{max} can be set to the maximum legal speed in the system area. Then, we form a K-anonymous cloaked area A as a minimum bounding rectangle of U and the adjusted location region of the $K-1$ peers (Line 16 in Algorithm 2). If A does not satisfy the minimum area privacy requirement, we extend A as in the original *cloaked area step* in Algorithm 1 (Line 17 in Algorithm 2).

4.3 Peer-to-peer spatial cloaking in a partitioned network

Since we consider a highly ad hoc mobile environment, where no fixed communication infrastructure or centralized/distributed servers are available, mobile users can only communicate with each other via multi-hop peer-to-peer routing. Due to user mobility, mobile users may be partitioned into disjoint networks. When a network partition takes place, a mobile user can only communicate with other peers residing in her network partition. If a user's required anonymity level is larger than the number of users residing in her network partition, the user cannot find enough peer location information to blur her location information; and thus, the user suffers from a network partition problem. In this section, we first discuss how to detect a network partition problem and two straightforward approaches to alleviate it, and then propose the *historical location scheme* to better alleviate the network partition problem.

We now discuss how to detect a network partition problem in the *peer search step* in Algorithm 1. Let $List_h$ be the set of peer location information that is found by the *peer search step* with a hop distance h. It is expected to find more peers as h increases, i.e., $|List_h| > |List_{h-1}|$. Thus, we know that a network partition takes place when $|List_h| = |List_{h-1}|$. It is important to note that we only need to record the size of $List_{h-1}$ without maintaining the peer location information. There

Algorithm 3 Historical Location Scheme

```
1: function HistoricalLocationScheme(User U, Tolerance tol_h)
2: //Step 1: Peer Search Step
3: if U detects |List_h| = |List_{h-1}| in the peer search step in Algorithm 1 then
4:    h_last ← h − 1
5:    h ← 1
6:    while |List| < U.K − 1 or h ≤ h_last do
7:       Send a request with U.K and tol_h to the peers within h hops
8:       List ← List∪ {the "uncertain" cached location information returned by the
         peers}
9:       h ← h + 1
10:   end while
11: end if
12: //Step 2: Cloaked Area Step
13: Adjust the uncertain peer location information in List
14: if |List| ≥ K − 1 then
15:   S ← {U} ∪ {the K − 1 peer nearest to U in List}
16: else
17:   S ← {U} ∪ List
18: end if
19: A ← a minimum bounding rectangle includes U, the peers with exact location
    information in S, and the adjusted location region of the peers with uncertain
    location information
20: Execute Lines 13 to 16 in Algorithm 1 to insure that A satisfies the minimum
    area privacy requirement
21: return A
```

are two straightforward approaches that a user can use to overcome the network partition problem. (1) After the user suffers from the network partition problem, she periodically performs the *peer search step* until she can find enough peer location information. (2) After the user cannot find an adequate number of peers in her network partition, she simply reduces her required K-anonymity level to the number of users residing in her network partition. However, these two straightforward approaches have drawbacks. The first approach would incur very long searching time while the second one requires the user to degrade her privacy protection. To this end, we propose the *historical location scheme* that enables the users to utilize their or other peers' cached information to better alleviate the network partition problem.

Algorithm Algorithm 3 depicts the pseudo code of the *historical location scheme*, and Fig. 3 gives an example to illustrate this scheme. Similar to the *information sharing scheme*, the user can control the staleness of the historical peer location information through a tolerance parameter tol_h. Thus, this scheme only uses the historical peer location information that was cached not earlier than $t_{now} - tol_h$. When U suffers from the network partition problem in the *peer search step* in Algorithm 1, i.e., $|List_h| = |List_{h-1}|$ and $List_{h-1} < K$, U sets $h_{last} = h - 1$ and executes the *historical*

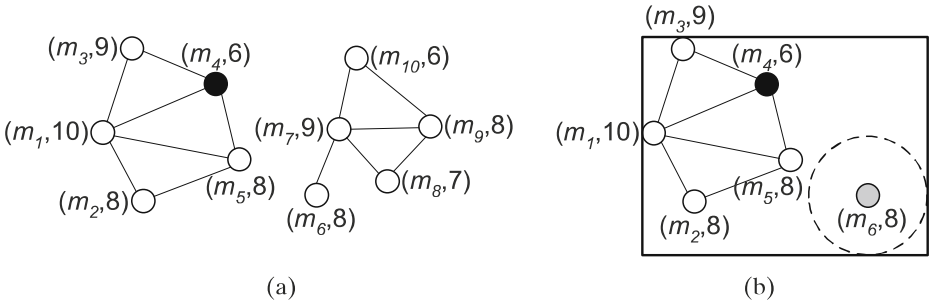

Fig. 3 Example of peer-to-peer spatial cloaking in a partitioned network. **a** Partitioned network. **b** Historical location scheme

location scheme. In general, the *historical location scheme* can be incorporated into our P2P spatial cloaking algorithm with the following modifications.

Step 1: ***Peer search step*** As *List* already stores the *exact* location information of the peers residing in U's network partition through the previous peer search process, U only needs to ask them to turn in their cached peer location information. Initially, U sends a request with a parameter tol_h and a hop distance $h = 1$ to her neighbor peers. Then, the neighbor peer sends the location information cached not earlier than $t_{now} - tol_h$ to U. U stores this cached location information in *List* and marks it as *uncertain*. When U receives duplicate peer location information, U only keeps the freshest one. Similar to the original *peer search step* in Algorithm 1, if U cannot find enough peer location information in *List* with a hop distance h, U increases h by one and rebroadcasts the request. This peer search process repeats until U finds enough peer location information or $h = h_{last}$ (Lines 6 to 10 in Algorithm 3). After U finishes the peer search process, she proceeds to the next step regardless of whether *List* stores enough peer location information.

Step 2: ***Cloaked area step*** The user U starts this step by adjusting the *uncertain* peer location information in *List* as in the *information sharing scheme*. If $|List| \geq U.K - 1$, U adds the $K - 1$ nearest peers in *List* and herself to S (Line 15 in Algorithm 3). For a peer P with *uncertain* location information, the distance between U and P is the maximum distance between U and the adjusted location region of P, i.e., $d_{max}(U, P) = d(U, P) + (t_{now} - t) \times v_{max}$. However, if $|List| \leq U.K$, we use the second straightforward approach to temporally reduce $U.K$ to $|List| + 1$ (Line 17 in Algorithm 3); and thus, U simply adds all peers in *List* and herself to S. Then, U forms a cloaked area A that includes the peers with the *exact* location information and the adjusted location region of the peers with the *uncertain* location information in S (Line 19 in Algorithm 3). If A does not satisfy the minimum area privacy requirement, we extend A as in the original *cloaked area step* in Algorithm 1 (Line 0 in Algorithm 3).

Example Figure 3 depicts an example of a partitioned network where the mobile users are partitioned into two disjoint networks. One network partition includes five

mobile users m_1 to m_5, while the other partition includes five mobile users m_6 to m_{10} (Fig. 3a). Each user is labeled with a pair of values, where the first value is her identity and the second value is her required anonymity level. In this example, m_4 (represented by a black circle) executes the P2P spatial cloaking algorithm to blur her location into a cloaked area. Since the number of users residing in m_4's network partition is less than her required anonymity level, i.e., $m_4.K = 6$, m_4 suffers from the network partition problem. After m_4 performs the peer search process with a hop distance of $h = 3$, she detects the network partition problem because $|List_2| = |List_3|$ (i.e., $h_{last} = 2$). We assume that m_5 caches the location information of m_6. Thus, m_4 can find enough peer location information after she enlists the peers within two hop distance for help to turn in their cached location information. *List* stores the location information of five peers, i.e., $List = \{m_1, m_2, m_3, m_5, \underline{m_6}\}$, where the peer with *uncertain* location information is underlined. Then, m_4 adjusts m_6's location information, and the adjusted location region of m_6 is represented by a dotted circle, as depicted in Fig. 3b. The cloaked area A of m_4 that is represented by a rectangle includes m_4, the peers with *exact* location information in *List*, i.e., m_1, m_2, m_3, and m_5, and the adjusted location region of m_6 (i.e., the dotted circle). We assume that A satisfies m_4's minimum area privacy requirement, so A is returned to m_4 as her cloaked area.

4.4 Cloaked area adjustment scheme

As discussed in Section 4.1, our P2P spatial cloaking selects the nearest peers of a mobile user to form her cloaked area A. This peer selection process may pose a privacy breach that the query issuer tends to be the closest to the center of A, i.e., the "center-of-cloaked-area" privacy attack [10, 15, 27]. Such a privacy breach may give more information to an adversary to infer the query issuer's actual location. To prevent this privacy breach, we should adjust A such that the probability of the query issuer being the closest to the center of A is $1/K$. To this end, we propose the *cloaked area adjustment scheme* that prevents the "center-of-cloaked-area" privacy attack if the user knows the exact location information of the peers residing in A. In case that the user only knows the historical location information of some peers residing in A, the *cloaked area adjustment scheme* can still alleviate the "center-of-cloaked-area" privacy attack. We will evaluate the resistance of our *cloaked area adjustment scheme* to this privacy attack through extensive experiments in Section 6.1.

Algorithm Algorithm 4 gives the pseudo code of the *cloaked area adjustment scheme*. This algorithm is called by Algorithm 1 after executing Line 12; thus, we add the statement "$A \leftarrow$ CloakedAreaAdjustment(U, S, A)" after Line 12 in Algorithm 1. The inputs to the algorithm include a user U, a set of users S selected by the *cloaked area step* in Algorithm 1, and a minimum bounding box of the users in S, A. A is represented by its bottom-left vertex (x_s, y_s) and top-right vertex (x_e, y_e), and the center of A is $C_A = ((x_s + x_e)/2, (y_s + y_e)/2)$. The output of this algorithm is an adjusted cloaked area A' that is represented by its bottom-left vertex (x'_s, y'_s) and top-right vertex (x'_e, y'_e), and C'_A is the center of A'. Initially, we set $A' = A$ (Line 2 in Algorithm 4). For the peer with historical location information in S, we consider the center of her adjusted location region as her location. We start this algorithm by randomly selecting a user P in S and finding the nearest user P_C in S to the center

Algorithm 4 Cloaked Area Adjustment Scheme

1: **function** CloakedAreaAdjustment(*User* U, *UserSet* S, *Area* A)
2: $C'_A \leftarrow C_A$; $A' = \{(x'_s, y'_s), (x'_e, y'_e)\} \leftarrow A = \{(x_s, y_s), (x_e, y_e)\}$
3: $P_C \leftarrow$ the nearest user in S to C_A
4: $P \leftarrow$ a randomly selected user in S
5: **if** $P \neq P_C$ **then**
6: //***Step 1: Center Adjustment Step***
7: $P_N \leftarrow$ the nearest user in S to P
8: $M \leftarrow$ the intersection point between line $\overline{PC_A}$ and the circle centered at P with a radius of $d(P, P_N)/2$
9: $R \leftarrow$ a random value within a range of $(d(M, C_A), d(P, C_A)]$
10: $C'_A \leftarrow \left(\frac{[d(P,C_A)-R]\times C_A.x+R\times P.x}{d(P,C_A)}, \frac{[d(P,C_A)-R]\times C_A.y+R\times P.y}{d(P,C_A)}\right)$
11: //***Step 2: Area Adjustment Step***
12: $\Delta_x \leftarrow |C'_A.x - C_A.x|$
13: $\Delta_y \leftarrow |C'_A.y - C_A.y|$
14: **if** $C'_A.x < C_A.x$ **then** $x'_s \leftarrow x_s - 2\Delta_x$; **else** $x'_e \leftarrow x_e + 2\Delta_x$
15: **if** $C'_A.y < C_A.y$ **then** $y'_s \leftarrow y_s - 2\Delta_y$; **else** $y'_e \leftarrow y_e + 2\Delta_y$
16: **end if**
17: **return** A'

of A, C_A. If $P = P_C$, we simply return the original A to Algorithm 1; otherwise, we have to adjust A by the following two main steps.

Step 1: ***Center adjustment step*** Figure 4a illustrates this step where the gray and black squares represent the center of the input A (C_A) and the center of the adjusted A' (C'_A), respectively. The user U starts this step by finding the nearest user P_N in S to P. Then, U computes the intersection point M between line $\overline{PC_A}$ and the circle centered at P with a radius of $d(P, P_N)/2$ (Line 8 in Algorithm 4). To guarantee that P will be the closest one to C'_A,

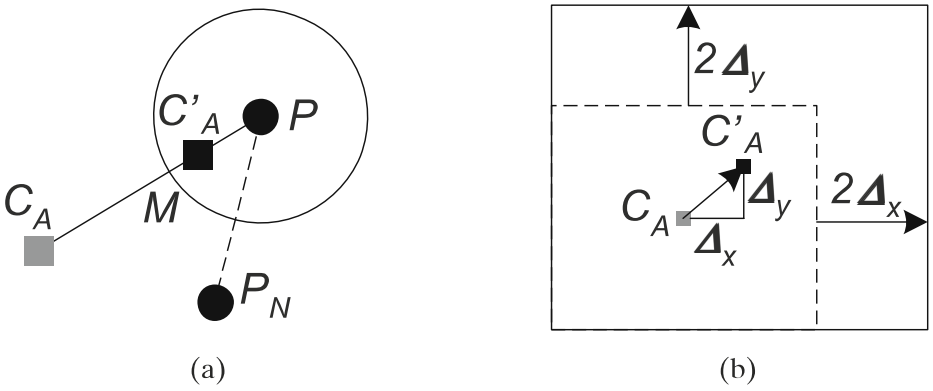

Fig. 4 Examples of the cloaked area adjustment scheme. **a** Center adjustment. **b** Cloaked area adjustment

C_A has to be moved towards P by a distance of greater than $d(M, C_A)$. We can set the upper bound of the adjustment distance to $d(P, C_A)$. If the adjustment distance is $d(P, C_A)$, C'_A will be located at P. To avoid revealing any location information of the user residing in A, we randomly select a value R within a range $(d(M, C_A), d(P, C_A)]$ (Line 9 in Algorithm 4). Then, C_A is moved towards P by a distance of R; hence, the adjusted center C'_A is:

$$\left(\frac{[d(P, C_A) - R] \times C_A.x + R \times P.x}{d(P, C_A)}, \frac{[d(P, C_A) - R] \times C_A.y + R \times P.y}{d(P, C_A)}\right).$$

Step 2: ***Area adjustment step*** Figure 4b illustrates this step where the solid and dotted rectangles represent the adjusted A' and the input A, respectively. After the user U determines the center of the adjusted A, C'_A, we adjust A such that C'_A is the center of A'. To form A', we can determine the difference between the coordinates of C_A and C'_A, i.e., $\Delta_x = |C'_A.x - C_A.x|$ and $\Delta_y = |C'_A.y - C_A.y|$ (Lines 12 to 13 in Algorithm 4). Then, we extend the nearest vertical edge of A to C'_A by a distance of $2\Delta_x$, and the nearest horizontal edge of A to C'_A by a distance of $2\Delta_y$. Mathematically, we can compute the coordinates of A' by using the equations given in Lines 14 to 15 in Algorithm 4.

5 Anonymous location-based services

To enable location-based database servers to support privacy-aware location-based queries with cloaked areas rather than with exact location points, the database server needs to be equipped with a privacy-aware query processor [5, 14, 15, 19]. In this paper, we adopt the work [14] as our privacy-aware query processor because it minimizes the communication overhead of sending a minimal answer set from the database server to the user while guaranteeing the user can get the exact answer. When a user wants to issue a privacy-aware location-based queries, she executes our P2P spatial cloaking algorithm to blur her location into a cloaked area. Then, the user sends the query along with the cloaked area to the location-based database server. After the privacy-aware query processor computes a minimal answer set that includes the exact answer to the user, the database server sends the answer set to the user. Finally, the user computes the exact answer from the answer set. The smaller the cloaked area, the smaller the answer set will be returned to the user. However, the user may need to relax her privacy requirements to achieve a smaller cloaked area. Thus, a trade-off between the user privacy protection and the quality of services can be achieved.

Figure 5 illustrates the privacy-aware nearest-neighbor query processing. Figure 5a depicts the data objects, e.g., gas stations, stored at the server side. There are 32 data objects T_1 to T_{32} represented by black circles, the shaded area represents the cloaked area of the mobile user who issued the query. For clarification, the actual mobile user location is plotted in Fig. 5 as a black square inside the cloaked area A, i.e., the shaded area. However, such information is neither stored at the server side nor revealed to the server. The privacy-aware query processor determines a minimal answer set that includes the nearest object to every point within A. It has been proved that the minimal answer set includes all objects within A and the nearest object to

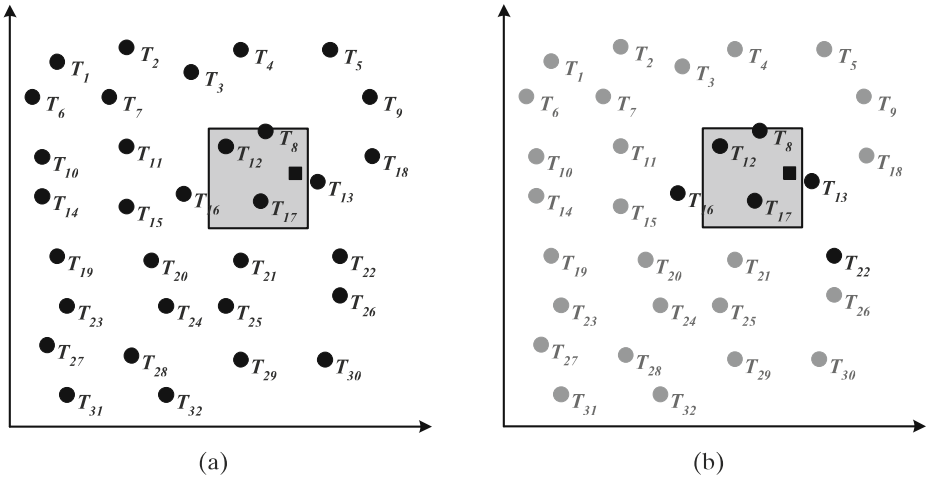

Fig. 5 Privacy-aware nearest-neighbor query processing. **a** Server side. **b** Client side

every point of each edge of A [14]. In this example, the server returns the answer set that includes six objects, i.e., T_8, T_{12}, T_{13}, T_{16}, T_{17}, and T_{22}, to the user. Then, the user computes the exact answer, i.e., T_{13}, from the answer set. The algorithmic detail of the privacy-aware query processor is beyond the scope of this paper. Interested readers are referred to [14] for more details.

6 Experimental results

In this section, we evaluate the performance of our P2P spatial cloaking algorithm (denoted as P2P) with the three key features, *information sharing scheme* (denoted as IS), *historical location scheme* (denoted as HL), and *cloaked area adjustment scheme* (denoted as CA) through simulated experiments. Our P2P spatial cloaking algorithm (P2P) corresponds to the *on-demand* approach of the state-of-the-art P2P spatial cloaking algorithm [4]. We do not consider the *proactive* approach because its communication overhead is much higher than the *on-demand* approach. Table 1 summaries the features of our algorithms.

We evaluate our algorithms with respect to five important performance measures. (1) *Number of messages.* This measure gives the average number of messages incurred by our algorithms per each query. It indicates the network bandwidth consumption and the power consumption on user devices. (2) *Cloaked area size.*

Table 1 Summary of the features of our algorithms

	P2P	P2P-IS	P2P-IS-CA	P2P-IS-CA-HL
Peer-to-Peer Spatial Cloaking (P2P)	√	√	√	√
Information Sharing Scheme (IS)	×	√	√	√
Cloaked Area Adjustment Scheme (CA)	×	×	√	√
Historical Location Scheme (HL)	×	×	×	√

Table 2 Parameter settings

Parameters	Default values	Ranges
Number of users	200K	100K to 500K
Number of querying users	20K	10K to 50K
Number of data objects	20K	10K to 50K
Transmission range	[100, 200] meters	[100, 100] to [100, 300] meters
K-anonymity	[50, 100]	[50, 100] to [50, 300]
Minimum area A_{min}	0	1 to 5 km^2
tol_s	0	0 to 50 s
tol_h	50 seconds	–
Movement speed	[50, 70] miles per hour	–

This measure gives the average size of cloaked areas generated by our algorithms. The smaller the cloaked area, the more accurate the location is reported to the location-based database server; the thus, this measure can indicate the location utility of our algorithms. (3) *Answer set size.* This measure indicates the average number of objects included in answer sets returned by the location-based database server. It also indicates the communication overhead of sending the answer set from the database server to the user. (4) *Anonymization success rate.* This is a ratio of the number of times that the anonymization algorithm can find enough peer location information to satisfy the user's K-anonymity privacy requirement to the total number of queries. (5) *Privacy attack probability.* This measure gives the resilience of our algorithms to the "center-of-cloaked-area" privacy attack by measuring the probability of the nearest user to the center of a cloaked area (U_c) being the actual query issuer, i.e., P(U_c is the query issuer).

We use a networked generator to generate moving objects on the road map of Hennepin County, Minnesota, USA. The road map consists of 57,020 edges and 42,135 vertices. Unless mentioned otherwise, all our experiments consider 100,000 mobile users in which 10% of them are randomly selected to issue nearest-neighbor queries (e.g., "*where is my nearest gas station*") at each time instance. The mobile users are moving at speeds between 50 and 70 miles per hour. Their required K-anonymity levels are uniformly selected from a range [50, 100], while their required minimum areas are set to zero. There are 20,000 data objects that are uniformly distributed within the underlying road map. The default uncertainty tolerance for the *information sharing scheme* (tol_s) and *historical location scheme* (tol_h) is set to zero and 50 seconds, respectively. We consider a heterogeneous network environment where the transmission range of each user is uniformly selected within a range [100, 200] meters. Table 2 summarizes the parameter settings.

6.1 Anonymization strength

Figure 6 depicts the resilience of our algorithms to the "center-of-cloaked-area" privacy attack with respect to varying (a) the K-anonymity level from 100 to 300, (b) the number of users from 100,000 to 500,000, (c) the transmission range from [100, 100] to [100, 300] meters, and (d) the uncertainty tolerance for the *information sharing scheme*, i.e., tol_s, from 0 to 50 seconds. The default K-anonymity level for (b)-(d) is 100. The results of P2P-IS-CA-HL, P2P-IS-CA, P2P-IS, and P2P are represented by black, gray, light gray, and white bars, respectively. The pattern

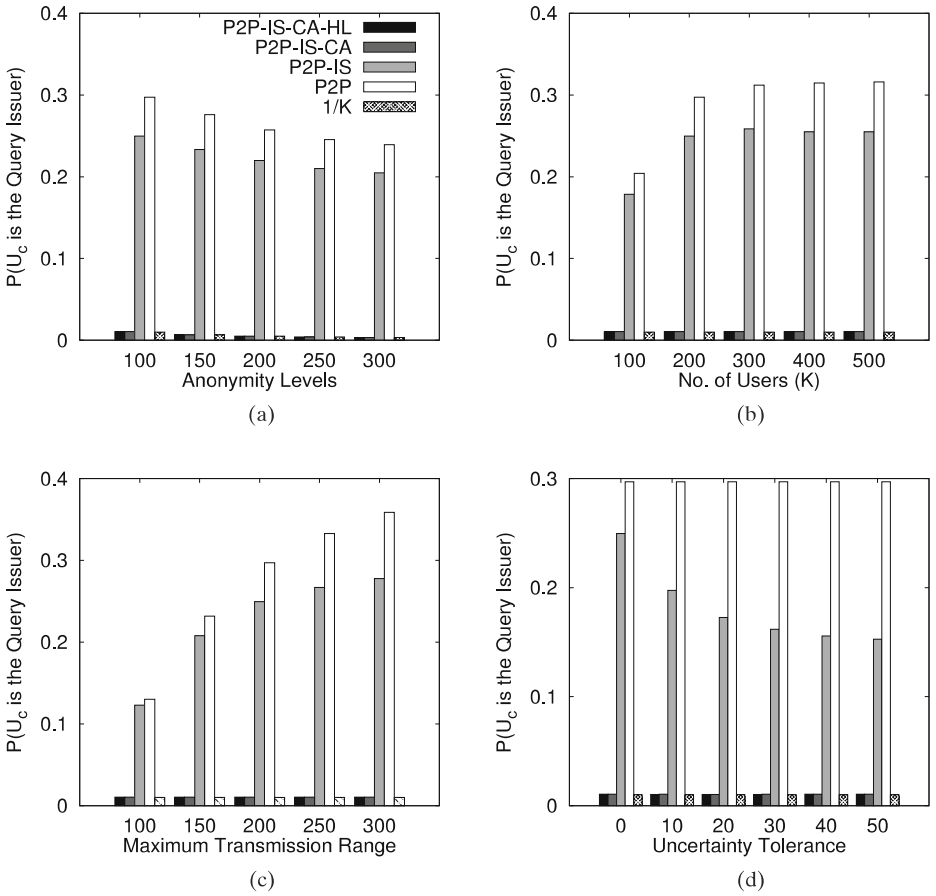

Fig. 6 "Center-of-cloaked-area" privacy attack. **a** K-anonymity. **b** Number of users. **c** Transmission range. **d** Uncertainty tolerance (tol_s)

bars represent the value of $1/K$ which indicates the ideal probability of the actual query issuer being the nearest user to the center of the cloaked area U_c, i.e., P(U_c is the query issuer). Although P2P is more vulnerable to the "center-of-cloaked-area" attack than P2P-IS (Fig. 6a–d), the probability of both P2P and P2P-IS are way above the ideal probability. The results show that our *cloaked area adjustment scheme* (CA) gives the probability of U_c being the actual query issuer is equal to or very close to $1/K$. Thus, our *cloaked area adjustment scheme* can effectively prevent the "center-of-cloaked-area" privacy attack.

6.2 Scalability

In this section, we evaluate the scalability of our algorithms with respect to large numbers of querying users, large numbers of users, and large numbers of data objects.

Figure 7 depicts the performance of our algorithms with respect to varying the number of querying users from 10,000 to 50,000 users, i.e., from 5% to 25% of

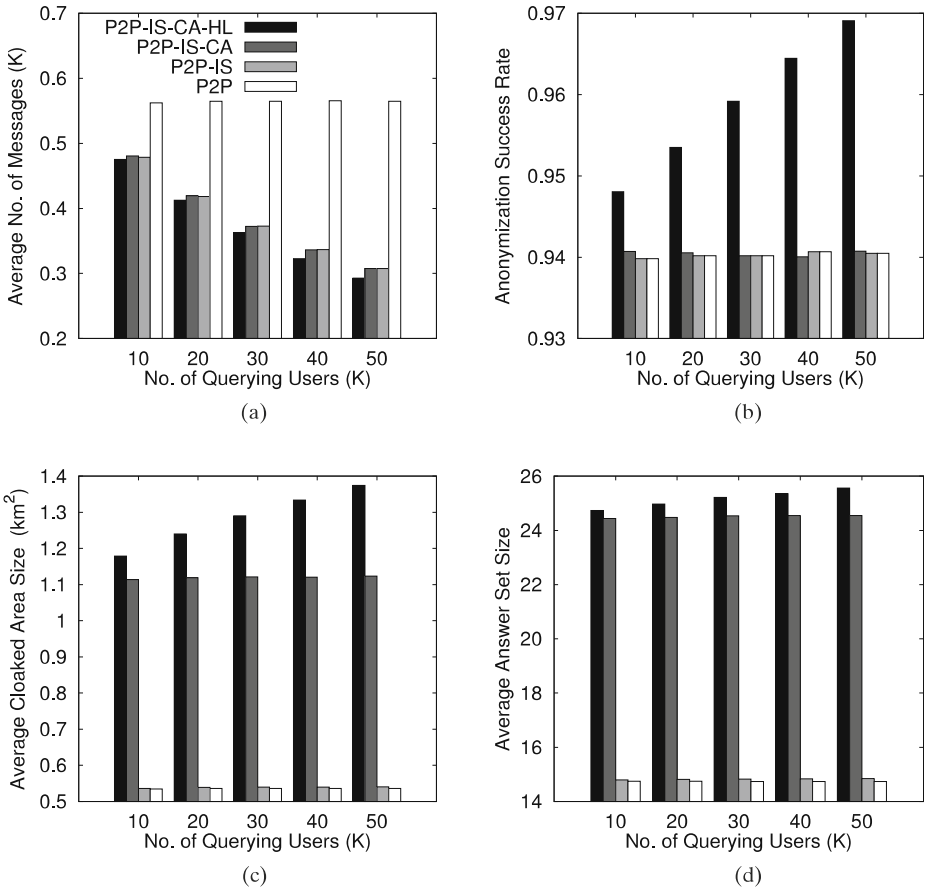

Fig. 7 Number of querying users. **a** Communication overhead. **b** Anonymization success rate. **c** Cloaked area size. **d** Answer set size

200,000 users. The performance of P2P is not affected by the number of querying users because there is no information sharing among users in P2P. The results show that our *information sharing scheme* (i.e., P2P-IS, P2P-IS-CA, and P2P-IS-CA-HL) effectively reduces communication overhead as there are more querying users (Fig. 7a). The main reason is that when the number of querying users increases, there is a higher chance for a user to obtain enough peer location information from her neighbor peers without searching the network. Likewise, when a user encounters a network partition problem, she is more likely to get enough peer location information from other peers residing in her network partition, as there are more querying users. Thus, the anonymization success rate of P2P-IS-CA-HL improves with more querying users (Fig. 7b). Since P2P-IS-CA-HL requires the users to adjust historical peer location to capture location uncertainty, they get larger cloaked areas than P2P-IS-CA (Fig. 7c). Processing such larger cloaked areas at the database server results in larger answer sets (Fig. 7d). It is important to note that although our algorithms with the *cloaked area adjustment scheme*, i.e., P2P-IS-CA

and P2P-IS-CA-HL, result in larger cloaked areas and answer sets than P2P and P2P-IS, P2P-IS-CA and P2P-IS-CA-HL are free from the "center-of-cloaked-area" privacy attack.

Figure 8 depicts the scalability of our algorithms with respect to increasing the number of users from 100,000 to 500,000. The results show that the performance of all algorithms gets better when there are more users. In a denser network, the user can find enough peer information to blur her location with a smaller hop distance, i.e., a smaller searching area; and hence, the communication overhead reduces (Fig. 8a). When the number of users increases, there are more users in a network partition. Thus, it is more likely that the user can find enough peer location information in her network partition, i.e., the anonymization success rate improves (Fig. 8b). Since the users can blur their locations into smaller cloaked areas in the denser network (Fig. 8c), the database server returns smaller answer sets to them (Fig. 8d).

Figure 9 depicts the scalability of our algorithms with respect to varying the number of data objects from 10,000 to 50,000. Since increasing the number of data

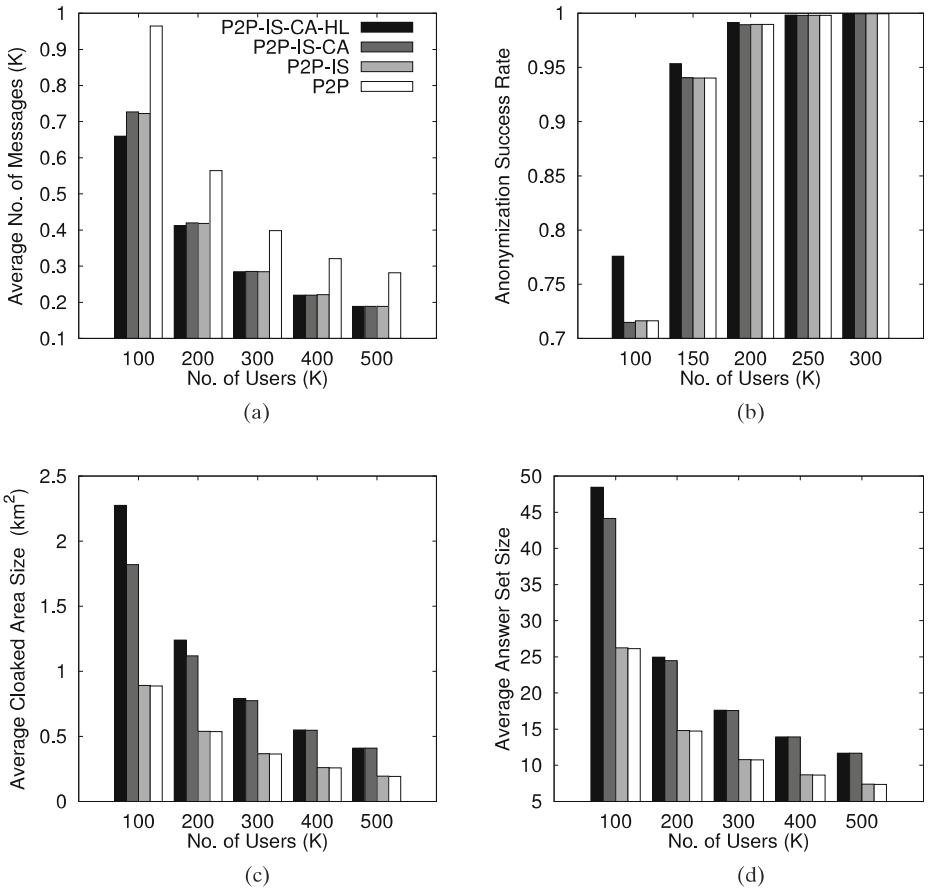

Fig. 8 Number of users. **a** Communication overhead. **b** Anonymization success rate. **c** Cloaked area size. **d** Answer set size

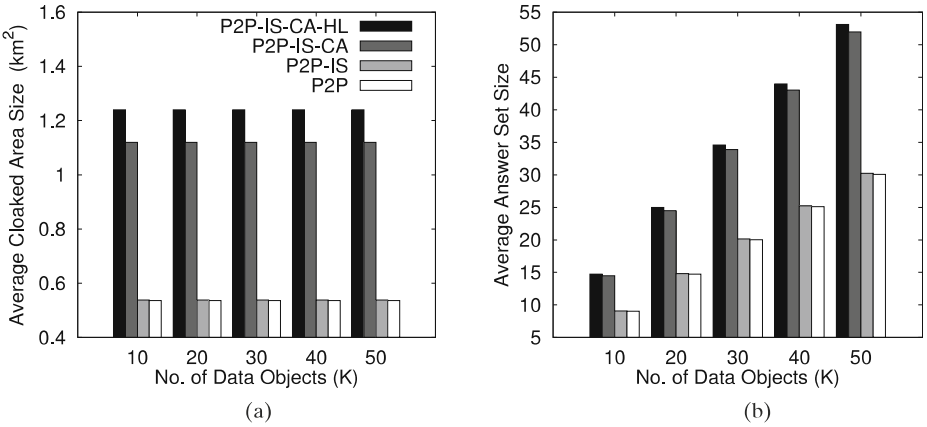

Fig. 9 Number of data objects. **a** Cloaked area size. **b** Answer set size

objects at the database server only affects the answer set size, the performance of the peer search process and the spatial cloaking process of all algorithms is not affected.Figure 9b gives that the answer set size of all algorithms linearly increases as there are more data objects.

6.3 Effect of privacy requirements

In this section, we evaluate the performance of our algorithms with respect to the user specified K-anonymity and minimum area A_{min} privacy requirements.

Figure 10 depicts the performance of our algorithms with the increase of the strictness of the K-anonymity privacy requirement from $[50, 100]$ and $[50, 300]$. It is expected that the performance of all algorithms becomes worse as the K-anonymity privacy requirement gets stricter. When K increases, the user needs to enlist more peers for help to gather enough peer location information, so the communication overhead gets higher (Fig. 10a). Our algorithms with the *information sharing scheme* (i.e., P2P-IS, P2P-IS-CA, and P2P-IS-CA-HL) perform better than P2P as K gets larger. Since the user with a stricter K-anonymity privacy requirement needs to gather more peer location information to blur her location, she is more likely to encounter the network partition problem (Fig. 10b). The results show that the user adopting our *historical location scheme* (i.e., P2P-IS-CA-HL) records a higher anonymization success rate than other algorithms. Figure 10c depicts that all algorithms generate larger cloaked areas to satisfy stricter privacy requirements. With larger cloaked areas, it is expected that the database server returns larger answer sets to the user (Fig. 10d).

Figure 11 gives the performance of our algorithms with respect to increasing the required minimum area A_{min} of cloaked areas from 1 to 5 km^2. In this experiment, we set the anonymity level to a smaller value, i.e., $K = 10$, so the results mainly depend on A_{min}. Varying A_{min} only affects the cloaked area size and the answer set size. It is interesting to see that when A_{min} is stricter than the K-anonymity privacy requirement, all algorithms give similar results in terms of cloaked area size and answer set size, as depicted in Fig. 11a, b, respectively. It is expected that all

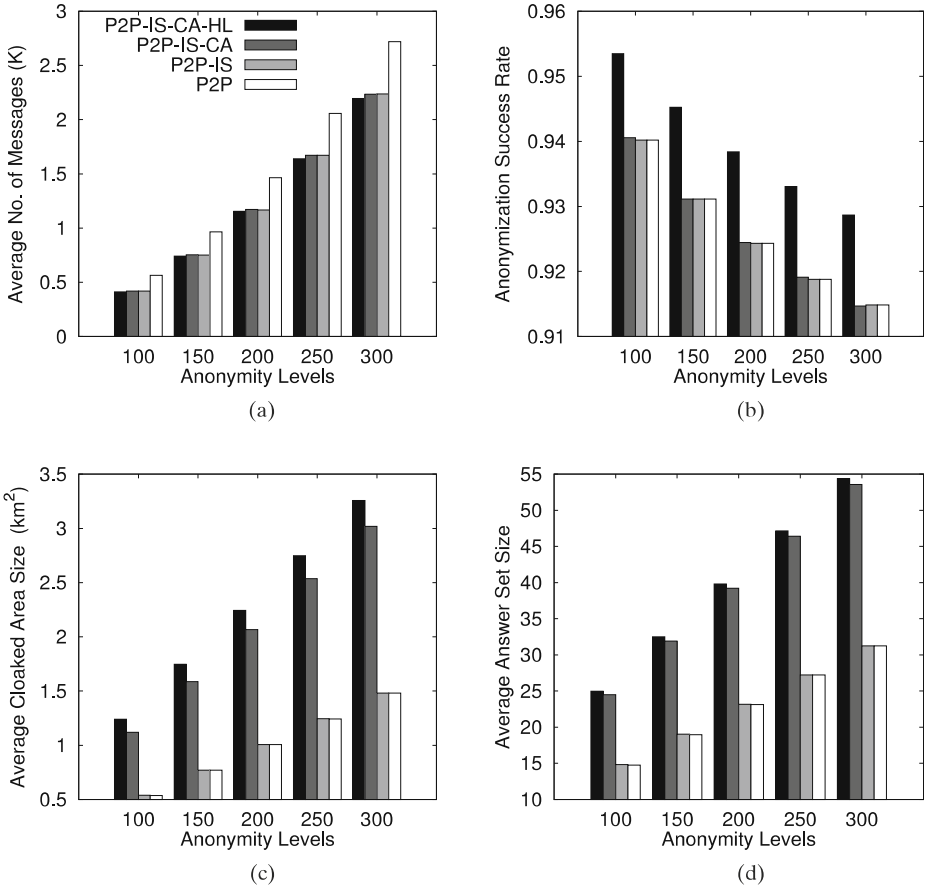

Fig. 10 *K*-anonymity privacy requirements. **a** Communication overhead. **b** Anonymization success rate. **c** Cloaked area size. **d** Answer set size

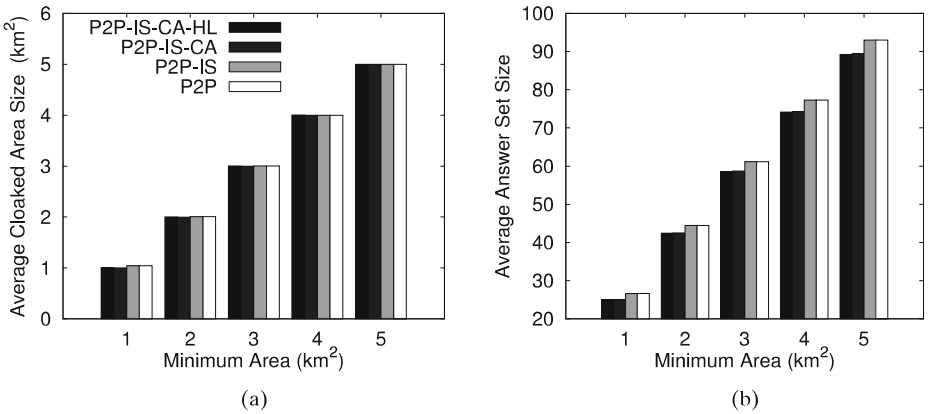

Fig. 11 Minimum area privacy requirements. **a** Cloaked area size. **b** Answer set size

algorithms generate larger cloaked areas as A_{min} gets stricter. When a cloaked area A gets larger, it is more likely that A includes a larger set of data objects and each edge of A has a larger set of nearest data objects. Thus, the answer set size increases as A_{min} gets stricter.

6.4 Effect of transmission range

Figure 12 depicts the performance of our algorithms respect to increasing the transmission range of user mobile devices from [100, 100] to [100, 300] meters. If the transmission range gets larger, the user can find enough peer location information within a smaller hop distance; and thus, the communication overhead reduces (Fig. 12a). As the transmission range gets larger, our algorithms can find K-anonymized cloaked areas for the users with stricter anonymity levels, so the anonymization success rate improves (Fig. 12b). However, the stricter the anonymity level, the lager the cloaked

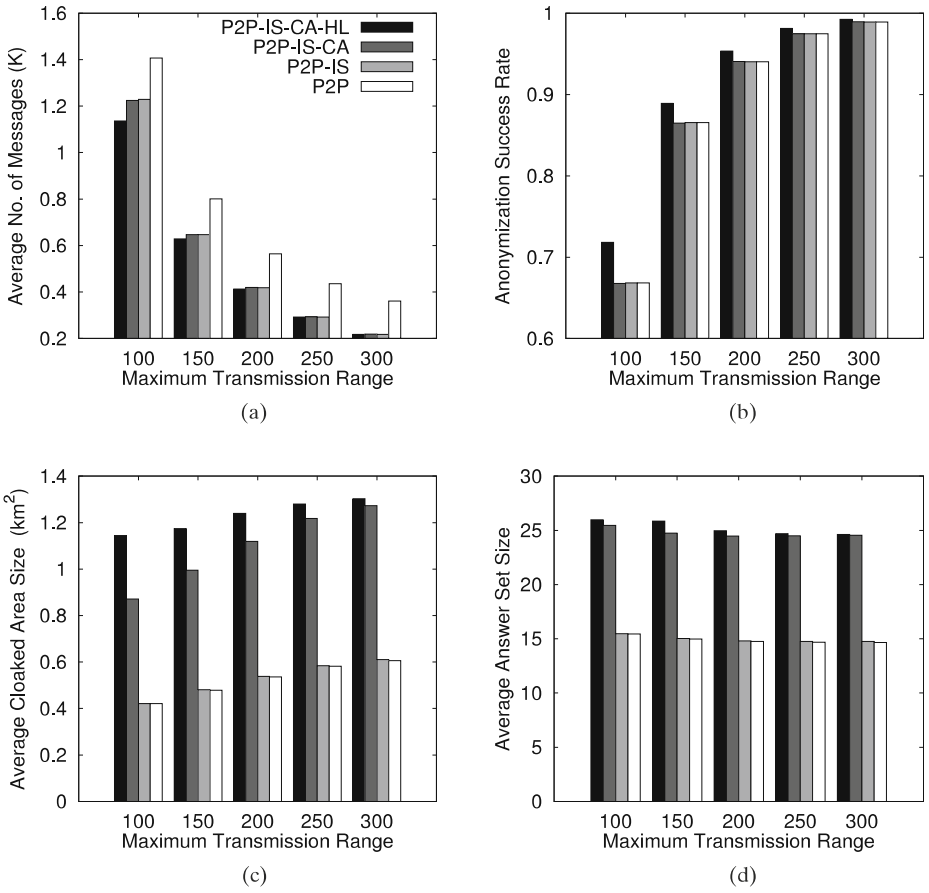

Fig. 12 Transmission range. **a** Communication overhead. **b** Anonymization success rate. **c** Cloaked area size. **d** Answer set size

area size is generated by our algorithms, so all algorithms generates larger cloaked areas as the transmission range increases (Fig. 12c). Since the increase of the cloaked area size is small, the answer set size is slightly affected (Fig. 12d).

6.5 Effect of uncertainty tolerance

Figure 13 gives the performance of our algorithms respect with to varying the user uncertainty tolerance level for the *information sharing scheme*, i.e., tol_s, from 0 to 50 seconds. Since P2P does not support information sharing among users, varying tol_s does not affect its performance. When the user is willing to utilize staler peer location information, it is easier for her to obtain enough peer location information from her neighbor peers without searching the network. Thus, the communication overhead of the algorithms with the *information sharing scheme* (i.e., P2P-IS, P2P-

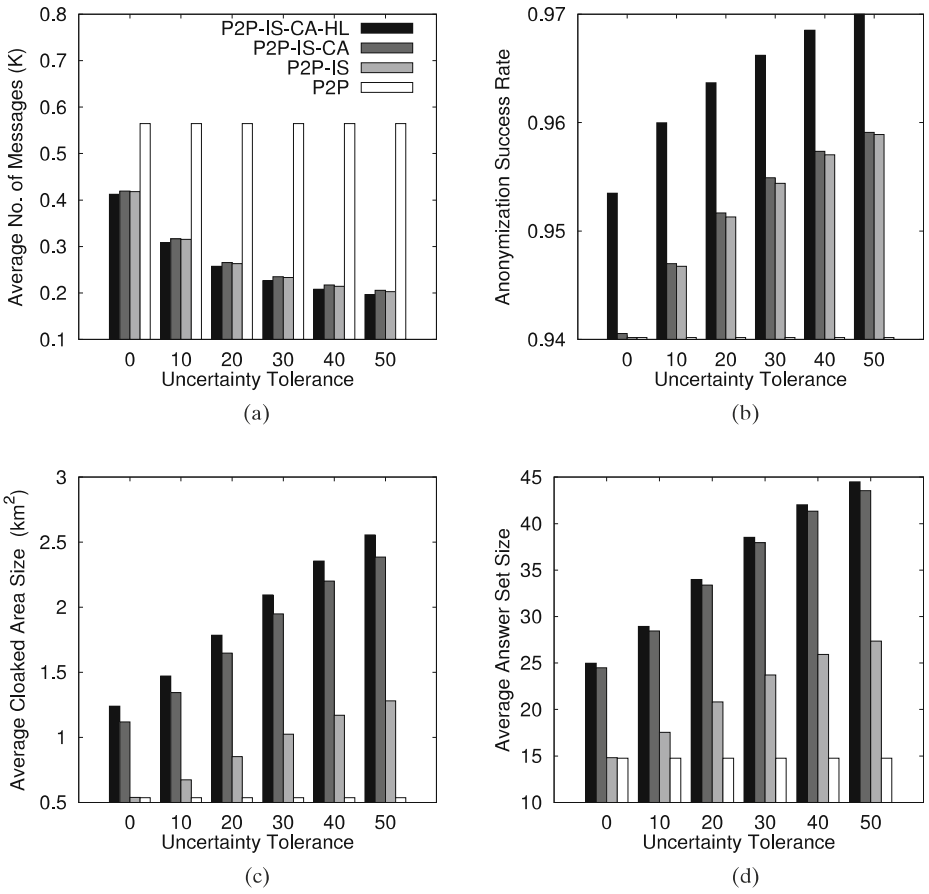

Fig. 13 Uncertainty tolerance for the information sharing scheme (tol_s). **a** Communication overhead. **b** Anonymization Success rate. **c** Cloaked area size. **d** Answer set size

IS-CA, and P2P-IS-CA-HL) reduces, as tol_s gets larger (Fig. 13a). Likewise, when the user accepts a larger tol_s, there is a higher chance for her to find enough peer location information within her network partition; and therefore, the user experiences a higher anonymization success rate as tol_s increases (Fig. 13b). With respect to cloaked area size, all algorithms generate larger cloaked areas, as tol_s increases (Fig. 13c). This is due to the fact that we need larger adjusted location regions for staler peer location information in order to capture its uncertainty. It is expected that the answer set size increases as the cloaked area gets larger (Fig. 13d).

7 Conclusion

In this paper, we present a peer-to-peer (P2P) spatial cloaking algorithm that enables mobile users to obtain location-based services without revealing their exact location information. Our P2P spatial cloaking algorithm is designed for mobile P2P environments in which no fixed communication infrastructure or centralized/distributed servers are available. The main idea of our algorithm is that when a mobile user wants to obtain location-based services, she collaborates with other peers to blur her location into a cloaked area. Our algorithm guarantees the cloaked area satisfies the user specified K-anonymity and minimum area A_{min} privacy requirements, i.e., the cloaked area includes at least K users and has a size of at least A_{min}. To overcome the limitations of mobile P2P environments, e.g., user mobility, limited transmission range, scarce communication resources, multi-hop communication, and network partition problem, we propose three key features for our algorithm. (1) The *information sharing scheme* enables mobile users to share their gathered peer location information with nearby peers in order to reduce communication overhead. (2) The *historical location scheme* allows users to utilize historical peer location information cached by other peers to alleviate the network partition problem. (3) The *cloaked area adjustment scheme* aims to adjust a cloaked area to guarantee that the adjusted cloaked area is free from a "center-of-cloaked area" privacy attack. We evaluate our P2P spatial cloaking algorithm with the three key features through extensive experiments. The experimental results show that our algorithm is scalable and efficient while guaranteeing the user's location privacy protection.

References

1. Bamba B, Liu L, Pesti P, Wang T (2008) Supporting anonymous location queries in mobile environments with privacygrid. In: Proceedings of the international world wide web conference. WWW
2. Chow CY, Leong HV, Chan ATS (2007) Grococa: group-based peer-to-peer cooperative caching in mobile environment. IEEE J Sel Areas Commun: Special Issue on Peer-to-Peer Communications and Applications, J-SAC 25(1):179–191
3. Chow CY, Mokbel MF (2007) Enabling private continuous queries for revealed user locations. In: Proceedings of the international symposium on advances in spatial and temporal databases. SSTD
4. Chow CY, Mokbel MF, Liu X (2006) A peer-to-peer spatial cloaking algorithm for anonymous location-based services. In: Proceedings of the ACM symposium on advances in geographic information systems. GIS

5. Chow CY, Mokbel MF, Nap J, Nath S (2009) Evaluation of range nearest-neighbor queries with quality guarantee. In: Proceedings of the international symposium on advances in spatial and temporal databases. SSTD
6. Duckham M, Kulik L (2005) A formal model of obfuscation and negotiation for location privacy. In: Proceedings of the international conference on pervasive computing. PerCom
7. Gedik B, Liu L (2008) Protecting location privacy with personalized k-anonymity: architecture and algorithms. IEEE Trans Mob Comput, TMC 7(1):1–18
8. Ghinita G, Kalnis P, Khoshgozaran A, Shahabi C, Tan KL (2008) Private queries in location based services: anonymizers are not necessary. In: Proceedings of the ACM international conference on management of data. SIGMOD
9. Ghinita G, Kalnis P, Skiadopoulos S (2007) PrivÉ: anonymous location-based queries in distributed mobile systems. In: Proceedings of the international world wide web conference. WWW
10. Ghinita1 G, Kalnis P, Skiadopoulos S (2007) Mobihide: A mobile peer-to-peer system for anonymous location-based queries. In: Proceedings of the international symposium on advances in spatial and temporal databases. SSTD
11. Gruteser M, Grunwald D (2003) Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the international conference on mobile systems, applications, and services. MobiSys
12. Gruteser M, Schelle G, Jain A, Han R, Grunwald D (2003) Privacy-aware location sensor networks. In: Proceedings of the workshop on hot topics in operating systems. HotOS
13. Hashem T, Kulik L (2007) Safeguarding location privacy in wireless ad-hoc networks. In: Proceedings of the international conference on ubiquitous computing. UBICOMP
14. Hu H, Lee DL (2006) Range nearest-neighbor query. IEEE Trans Knowl Data Eng, TKDE 18(1):78–91
15. Kalnis P, Ghinita G, Mouratidis K, Papadias D (2007) Preventing location-based identity inference in anonymous spatial queries. IEEE Trans Knowl Data Eng, TKDE 19(12):1719–1733
16. Khoshgozaran A, Shahabi C (2007) Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In: Proceedings of the international symposium on advances in spatial and temporal databases. SSTD
17. Kido H, Yanagisawa Y, Satoh T (2005) An anonymous communication technique using dummies for location-based services. In: Proceedings of IEEE international conference on pervasive services. ICPS, pp 88–97
18. Ku WS, Zimmermann R, Wang H (2008) Location-based spatial query processing with data sharing in wireless broadcast environments. IEEE Trans Mob Comput, TMC 7(6):778–791
19. Mokbel MF, Chow CY, Aref WG (2006) The new casper: query processing for location services without compromising privacy. In: Proceedings of the international conference on very large data bases. VLDB
20. Pfitzmann A, Kohntopp M (2000) Anonymity, unobservability, and pseudonymity - a proposal for terminology. In: Proceedings of the workshop on design issues in anonymity and unobservability, pp 1–9
21. Sweeney L (2002) K-anonymity: a model for protecting privacy. Int J Uncertain Fuzziness Knowl-Based Syst 10(5):557–570
22. Wu W, Tan KL (2006) Global cache management in nonuniform mobile broadcast. In: Proceedings of the international conference on mobile data management. MDM
23. Xu T, Cai Y (2007) Location anonymity in continuous location-based services. In: Proceedings of the ACM symposium on advances in geographic information systems. GIS
24. Xu T, Cai Y (2008) Exploring historical location data for anonymity preservation in location-based services. In: Proceedings of the international conference of the computer and communications societies. INFOCOM
25. Yiu ML, Ghinita G, Jensen CS, Kalnis P (2009) Outsourcing search services on private spatial data. In: Proceedings of the international conference on data engineering. ICDE
26. Yiu ML, Jensen C, Huang X, Lu H (2008) Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: Proceedings of the international conference on data engineering. ICDE
27. Zhang C, Huang Y (2009) Cloaking locations for anonymous location based services: a hybrid approach. GeoInformatica 13(2):159–182

Chi-Yin Chow (B.A., The Hong Kong Polytechnic University, 2002, M.Phil., The Hong Kong Polytechnic University, 2005, M.S., University of Minnesota, 2008) is a Ph.D. candidate in the Department of Computer Science and Engineering, University of Minnesota - Twin Cities. His main research interests are in spatial and spatio-temporal databases, mobile data management, wireless sensor networks, mobile peer-to-peer computing, location-based services, and data privacy. He was awarded the best paper award of the 10th International Conference of Mobile Data Management (MDM 2009). He was an intern at the IBM Thomas J. Watson Research Center during the summer of 2008. He is a student member of the ACM and the IEEE.

Mohamed F. Mokbel (Ph.D., Purdue University, 2005, MS, B.Sc., Alexandria University, 1999, 1996) is an assistant professor in the Department of Computer Science and Engineering, University of Minnesota. His main research interests focus on advancing the state of the art in the design and implementation of database engines to cope with the requirements of emerging applications (e.g., location-aware applications and sensor networks). Mohamed was the co-chair of the first and second workshops on privacy-aware location-based mobile services, PALMS, 2007 (Mannheim, Germany) and 2008 (Beijing, China). He was also the PC co-chair for the ACM GIS 2008 conference. Mokbel has spent the summers of 2006 and 2008 as a visiting researcher at Hong Kong Polytechnic University and Microsoft Research, respectively. He is a member of ACM and IEEE.

Xuan Liu is a research staff member at IBM T.J. Watson Research Center. Her research interests include privacy-aware mobile services, spatial databases, cloud computing, and knowledge management. Her research work has been incorporated into IBM WebSphere product family, customer solution offerings, and IBM Web services toolkit. Xuan received her PhD in Computer Science from University Of Minnesota, Minneapolis, USA. She is a member of IEEE, IEEE Computer Society and ACM.