On preserving Location Privacy with Split Cloaking

Chien-Ping Wu   Jiun-Long Huang

National Chiao Tung University

Hsinchu, Taiwan, ROC

{cpwu, jlhuang}@cs.nctu.edu.tw

1896

Abstract

With the progress of mobile devices and positioning technologies, the location-based services (LBSs) have emerged as one of the killer applications. And preserving personal location information becomes a very important issue for the LBS users. In this paper, we propose a new concept of k-anonymity to preserve users' location privacies. We introduce a novel cloaking algorithm, called Split-Cloak, to achieve higher k-anonymity level and obtain fewer candidate answers for the LBS clients. Our model enables the mobile users to send their location-based queries without revealing their exact locations and just need few support from LBS providers. The cloaking algorithm can achieve not only high k-anonymity level but also obtain smaller answer size. Our experiments show that the Split-Cloak algorithm can preserve users' location privacy well.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Nowadays, mobile devices, such as cell phones, PDAs, even laptops, become common equipments in humans daily lives. The popularity of these communication devices with embedded Global Positioning Services (GPS) are increasing significantly. This circumstances trigger the location-based services (LBSs) being one of the most important applications. Imagining that a tourist who wants to know the nearest scene spot, or a passenger who wants to get a taxi as soon as possible, all he can do is revealing his location to the location-based service servers and then receiving the required information in return. Moreover, LBSs are needed in many circumstances to facilitate our living. Just imaging again that a person who want to get to the nearest hotel in a foreign place; a driver who wants to get to the nearest gas station; or a gastronomer who want to know all the restaurants' information nearby. In these situations, users must provide their locations in order to obtain the information they queried. While LBSs have shown to be valuable to users' daily life, on the other hand, they also expose extraordinary threats to user's privacy. If the location information not be protected well, a malicious entity who retrieves the user's location information will become a tremendous threat. Thus, protecting location information is undoubted an important issue.

Recently, there are many researches dealing with preserving location privacy. [1], [2], [3], [4], [5] , [6], [7], [8], [9], [10], [11], [12]. These researches can be classified: one is *trusted-anonymizer*-based, the other is *client*-based. In the former, a user submits his/her query with exact location to the LBS servers via thrusted anonymizer, which is called the *trusted third party*. After the trusted third party receives users' exact locations, it blurs the location into a region, which is called the *cloak region*, to conceal the exact point from the LBS servers. LBS servers will then generate "candidate" answers back to the users. *K-anonymity* approach is commonly used in the literature. The second approach uses no trusted third party. Therefore, clients are responsible to conceal their locations from LBS servers by themselves. Peer-to-peer approach [3] and dummy approach [7] are introduced in this part.

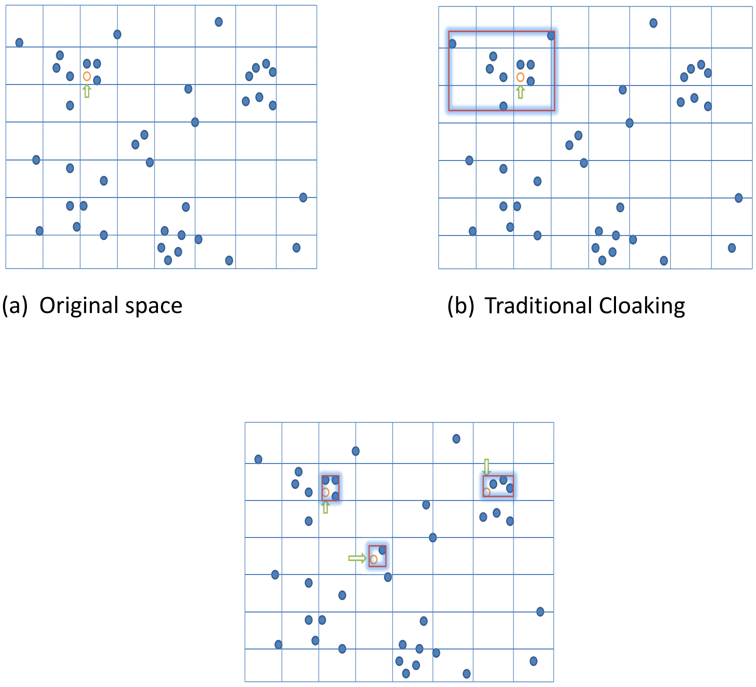
1.1 K-anonymity

K-anonymity was first introduced by L. Sweeney in [11]. It was first used to protect users profiles in database system. Afterward, this concept was applied to protect location data. In the context of LBSs and mobile users, location k-anonymity refers to k-anonymous usage of location information. We can consider a location as k-anonymous if and only if the location information sent from a mobile user to LBSs is indistinguishable from other k-1 locations which are sent from other different mobile users. It means if an adversary gets a location information, she will have uncertainty in matching a specific mobile user with that location. Ideally, when the value of k increases, it will provide better privacy for users.

1.2 Contributions of this Paper

In this paper, we propose a customizable k-anonymity model and a new concept of k-anonymity to preserve users' location privacies. Many approaches, such as [3] [4] [10], achieve k-anonymity by blurring one point into exact one region which contains at least k-1 other users. There exists a lethal problem: when the value of k is large, the cloak region should also enlarge to contain other k-1 points. This will make the LBS providers consuming more computational power to find out more candidate answers. Moreover, the query will fail if the value of k is too large just because it can't find enough k-1 other points to cloak. In order to solve this problem, we propose a new idea of split cloaking against traditional methods. When users issue LBS queries with large value of k, we will cloak them into more than one regions. Based on this concept, we split the original cloaked region which is large into several small regions. Figure 1.1 depicts this concept. Obviously, even if the user issues a query with a large value of k, we can split it into a set of smaller k*s* and still satisfy the k-anonymity constraint. There are a number of challenges when we split a large region into several small ones. The first challenge is that how many regions should we select such that they still satisfy the k-anonymity constraint. The second challenge is to determine each separate regions such that each region should have a smaller scope of area to decrease the return answer size from LBS providers.

We introduce a novel cloaking algorithm, called Split-Cloaking, to achieve higher k-anonymity



(a) Original space

(b) Traditional Cloaking

(c) Proposed distributed cloaking

Figure 1.1: Traditional methods of cloaking vs. distributed cloaking

level and obtain smaller candidate answer sets for the LBS clients. First, we address two different types of queries for LBS users, privacy-concerned and area-concerned. For privacy-concerned, users can determine their privacy level by setting the value of k in their query messages. For area-concerned, users can limit the area of cloaked regions to decrease the return answer size from the LBS providers. Second, we propose an optimization model to optimize the cloaked regions' area under the same privacy level or to maximize the privacy level when the cloaked regions' area is restricted within certain limits. Third, we introduce the potential users to participate in the cloaking process to avoid the failure of a query when the LBS queries are not enough. Finally, we construct a series of experiments on the performance of the cloaking algorithm under various conditions. Our experiments show that with Split-Cloaking we can achieve relative high k-anonymity level and small cloaked region area without making the query failure.

We organize this paper as follows: Section 2 introduces the related works, briefly describe their algorithms, system models, and indicate the drawbacks of each approach. Then we describe the motivation of our work in order to solve those problems. We also claim the preliminaries of our work, including assumptions and notations in this section. Our approach is described in Section 3, called *Split-Cloaking with Partial Time-dependency*. We will explain what the terms *Split* and *Partial* means in that section. Section 4 will show our evaluation results. Finally we conclude this paper in Section 5.

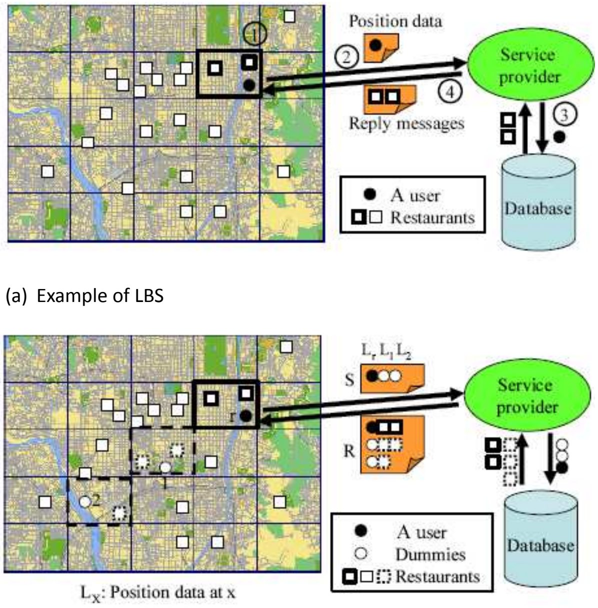
Chapter 2

Preliminaries

2.1 Related Works

There many approaches and also still many researches on preserving users' location privacy. [7] uses dummies to blank its true position out. The querying user will generate many dummies, which are fake positions of a query user, and then send all positions with just one valid. In this scenario, when the number of dummies increases, it will provide better privacy level for that user. Since the adversary will feel more confused about the true user's location, and must take more efforts to find the true position out. However, this approach encounters a serious problem: the LBS servers must take more efforts on processing the fake positions. This defect will strike when there are abundance of dummies. Thus, it will also cause low capabilities of the system in the same contention. Figure 2.1 depicts this drawback. Notice that in Figure 2.1(a), the return message only contains two restaurants, but in Figure 2.1(b), the return message contains five restaurants when two dummies are generated.

[6] applies a trusted server to conceal the location from LSB servers. After receiving users' locations, the trusted server uses Hilbert Curve to *encode/transform* the users true position to a transformed space. And so does the POIs (Point Of Interests). The LBS servers only know the transformed information about users' locations and POIs. Although this approach can conceal the real positions from the malicious entity (e.g., LBS servers), it will have strict constraints on this scenario. In most occasions, POIs should be owned by the LSB servers, not the third parties.In addition, cloaking is the most popular approach to conceal the users' exact locations. The concept of *k-anonymity* is applied here frequently. [11] first introduced the concept of *k-anonymity*. [3] [4] [5] [8] [9] [10] [12] use this concept on



(a) Example of LBS

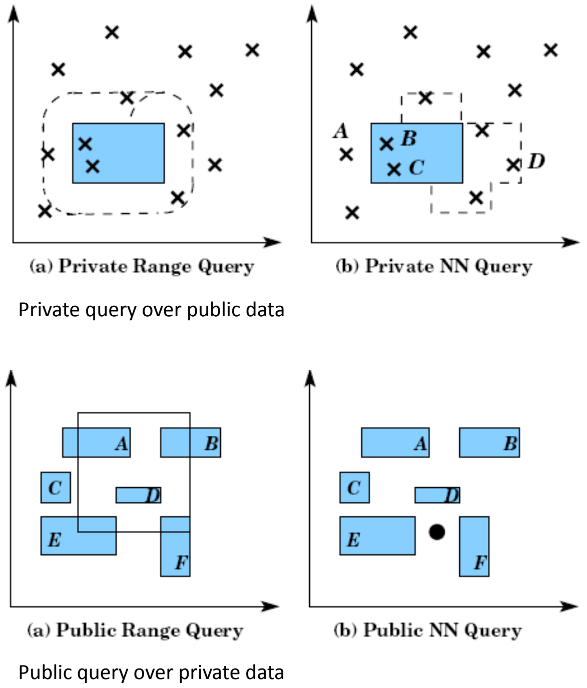
(b) Example of LBS using dummies

Figure 2.1: Example of LBS without and with dummies

location information. In [4] [9] [10] [12], system uses the trusted third party to anonymize users' location information. The anonymization server receives user's exact location, and then blurs it into a cloaked region. In this region, there are at least k users and the adversary can't distinguish a specific one from others. We call this location k-anonymity. However, [3] [5] use client-based structures. It means that the users have the responsibilities to hide their own positions. They must communicate with each other to find out who can help them to accomplish the cloaking process.

[9] addresses four novel query types: 1) private queries over public data, 2) public queries over private data, 3) private queries over private data, 4) public queries over public data. In LBS services, 1) is the most general query type. 2) and 3) is usually used for some special purposes, such as finding someone in a foreign area. Figure 2.2 depicts two of these query types. And then [10] proposes an architecture to support these queries: The New Casper. First, The New Casper propose a structure called pyramid to maintain users' location information. All users need to report their positions back to the location anonymizer and the location anonymizer will store their positions in the structure. Notice that users must update their positions when they leave their current grid to another. It means that the anonymizer must know all the users' location information even they don't send LBS queries. Thus, the load of the anonymizer will be heavy deservedly. Generating the cloaked region is just one of this paper's contributions. The other contribution is about the query processing.

[3] proposes a peer-to-peer approach to preserve user's location privacy. Unlike trusted-



(a) Private Range Query

(b) Private NN Query

Private query over public data

(a) Public Range Query

(b) Public NN Query

Public query over private data

Figure 2.2: Example of two query types

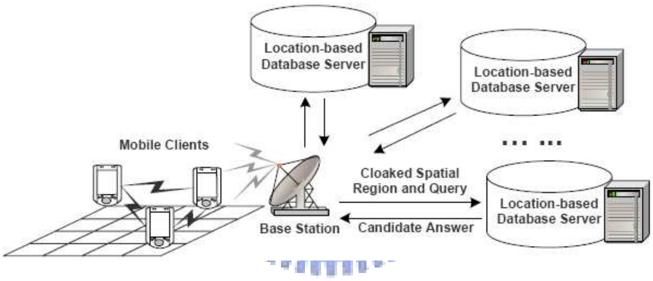


Figure 2.3: P2P system architecture

anonymizer-based architecture, users protect their location privacy without seeking help from any trusted third party. The architecture only involves in two components: clients and LBS providers (shows in Figure 2.3).

Ideally, all these approaches can reserve the users' location privacy, but, in fact, there are still many issues that they do not resolve. In client-based architecture, the user must trust other peers/clients other than the LBS servers because it should need help from other peers/clients to blur its location with cloaking algorithm. In many cases, the malicious one should be another user, not LBS servers. Thus, in real world, the client-based architecture may bear more threats from other users than the LBS servers. However, in trusted-anonymizer based architecture, the anonymizer takes charge of preserving users location privacy. After receiving users' location information, the anonymizer will cloak them into a region which satisfies the users' privacy profiles (k-anonymity). But according to the distribution of query

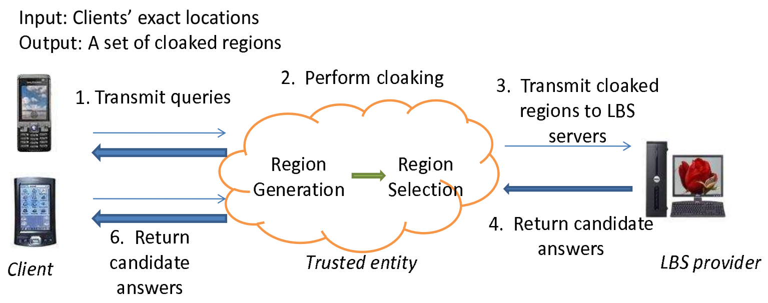

Input: Clients' exact locations
Output: A set of cloaked regions
1. Transmit queries
2. Perform cloaking
3. Transmit cloaked regions to LBS servers
Region Generation
Region Selection
4. Return candidate answers
6. Return candidate answers
Client
Trusted entity
LBS provider

Figure 2.4: System Model

points, the value of k may significantly small to ensure the success rate of the query. If the value of k is larger than some specific value (e.g., 5), the query maybe fail frequently. It means that users should choose a small value of k in order to make the query success. In other words, users privacy level will be low. Meanwhile, the malicious entity will have more chance to discover the true position of the user.

2.2 System Model

As shown in Figure 2.4, we have three components in our system: clients, the trusted entity, and LBS servers. Clients request for LBS services. They will transmit their exact locations to the trusted entity securely via secure wireless channels. After receiving the location information, the trusted entity performs cloaking algorithm to hide the exact location information from being known by untrusty entities, such as LBS servers. Then LBS servers will find candidate answers based on the cloaked regions and send them back to the trusted entity. Finally, the trusted entity will return the answers to the clients individually. We will have many benefits by using this model. First, clients need not to do any change. They just send their location information and anticipate that the answers will return in a few seconds. Second, LBS server just need to perform a preprocess to separate the cloaked regions into individual queries (because we split the cloaking region which should be a large area in original cloaking methods into several smaller ones).

2.3 Assumptions and Notations

In the real world, any user who using LBS services should be baleful. When we issue LSB queries, we can't release our true positions to them just for cloaking. Therefore, we apply *trusted anonymizer* based model in our system. Users must transmit their locations to the trusted anonymizer via secure wireless channels. The anonymizer will be responsible for blurring users' positions into a cloaked region. Before cloaking, we define the set of messages which are sent from users to the anonymizer as Q. We formally define the messages format in the set Q as follows :

$m_c \in Q :< ID, (X, Y), K, g >$

ID is the identity of the user who issuing this query, (X,Y) represents the user's exact position, K means the user's claim of its k-anonymity level. Finally, g is a system parameter that decides the degree of the global disperse. We will explain what global disperse means in the next section.

After receiving location information from querying users, the anonymizer will perform our cloaking algorithm (which will be presented in Section 3). And after cloaking process finishes, the anonymizer will generate "a set of regions". Let $M_i =< (xi_{lu}, yi_{lu}), (xi_{rd}, yi_{rd}) >$ represent the ith region in our cloaking approach, which $< (xi_{lu}, yi_{lu}), (xi_{rd}, yi_{rd}) >$ represent the MBR of that region (left-up and right-down points). And we denote A_i as the area of the MBR M(i) and A_{total} are denoted as the sum of those areas which are selected in our cloaking algorithm. We define the cloaked message format which will be sent to the LBS providers as follows :

$R =< (M_1, k_1), (M_2, k_2), ..., (M_n, k_n) >$, where $k_i, i = 1...n$, represents the regions' local k-anonymity. And we denote a subset C $\supseteq$ Q, where the set C contains the query messages which participate in R. Besides, we denote the set of messages which are sent by potential users as P. They will send their location information by requests of the anonymizer. Finally, $\forall m_c \in$ C, $k_1 + k_2 + ... + k_n \geq K$ must be satisfied.

2.4 Local and Global k-anonymity Constraints

To provide better privacy level, the following two properties should be considered.

Property 1. Local Dense: High local density can achieve the k-anonymity more easily and the cloaked region will be smaller, too. This property will provide better privacy level for those users in the same cloaked region.

Definition 1. We denote $LD(i) = k_i/A_i$, where k_i and A_i represent the k-anonymity level and the area of that region respectively. We say a cloaked region satisfies the property 1 if and only if $\{LD(i)|LD(i) > LD(i'), \forall i\}$, where $LD(i')$ represents the next extension of $LD(i)$. That means any expansion from current cloaked region will cause lower query density of that region.

Property 2. Global Disperse: Global disperse can increase the difficulty of a malicious entity to find out a specific user. Thus, it preserves user's location privacy better.

Definition 2. We say a cloaked message R satisfies global disperse if and only if $\{n|n \geq g, \forall m_c \in C\}$, where n is the number of regions in R.

Goal 1. High success rate for a query without compromising user's privacy: Larger k-anonymity can provide better location privacy level for users. The larger value of k can be satisfied, the higher privacy level can be achieved. Although larger k can provide better privacy level for users, it gets more difficult to find a cloaked region that contains enough other users.

Goal 2. Few update messages In many situations, users need to report their location information to the trusted anonymizer even if they do not send LBS queries. We propose a model to decrease the amount of this kind of messages.

Goal 3. Small cloaked regions' area: The cloaked regions' area can imply the candidate answer size. Smaller area implies that the return answer size will be smaller, too. In other words, a larger area will contain more candidate answers, and this will lead to an inefficient system.

The challenge is that: when we cloak the query points, how can we achieve the above properties even if the value of k is large. In addition, how can we select the appropriate regions that not only satisfy the users' k-anonymity but also ensure of the quality of return answer size (smaller region area leads to few candidate answers and no query will be omitted). Figure 2.5 illustrates the distribution of the query points. Figure 2.5(a) satisfies the property 2 (global disperse), but do not meet the property 1 (local dense). In contrast, Figure 2.5(b) satisfies the property 1, but property 2 is violated. Figure 2.5(e) is the best case that both of the two properties are accomplished.

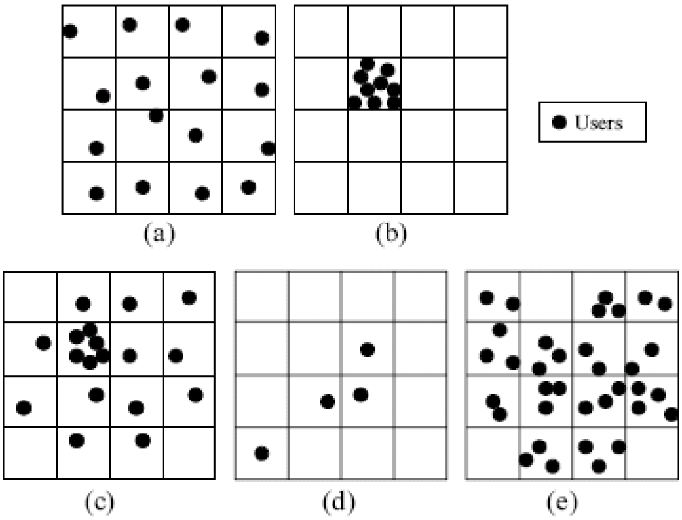


Figure 2.5: Example of user positions

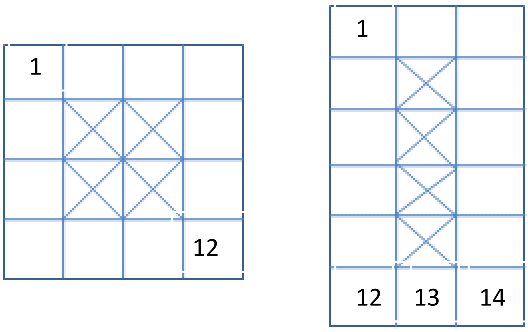
Chapter 3

The Split-Cloak Algorithm

We propose a new concept of cloaking algorithm here, called Split-Cloaking. we assume that the entire area is divided into grid spaces and anyone who issues a LBS request must locate in one of the grid. In our algorithm, there are two steps involved : region generation step and region selection step. In region generation step, we will find a set of cloaked regions, called candidate regions, which satisfy the property 1). In region selection step, we need to pick up enough region to satisfy the property 2). Here we propose an algorithm to determine which region should be selected to decrease the sum of those regions' area.

3.1 Region Generation: Split-Cloaking

There are two phases in the region generation step: 1) search phase, 2) split-cloaking phase. As shown in Algorithm 1, in the first, the anonymizer will select the earliest one (we called head) from all of the queries to avoid the delay time being too long, and then go into the search phase. In the search phase, the cloaking process have to contain more other query points to satisfy the property 1). Therefore, we search all the neighbor grids, then pick up the grid which contains the most points as a candidate cloaked region. If the density of that grid is higher than the original one, we merge it, and then continue seeking other grids . Notice that in order to avoid the cloaked region stretching too long, the next search will perform in different direction. (e.g. if the first search merges the right/left grid, the next search will search upward and downward.) Figure 3.1 shows that if the cloaked region both contain four grids, Figure 3.1(a) will have smaller region to search for the query answers than Figure 3.1(b). The search will stop untill the neighbor grids don't have higher density. Figure 3.2 depicts the



(a)

(b)

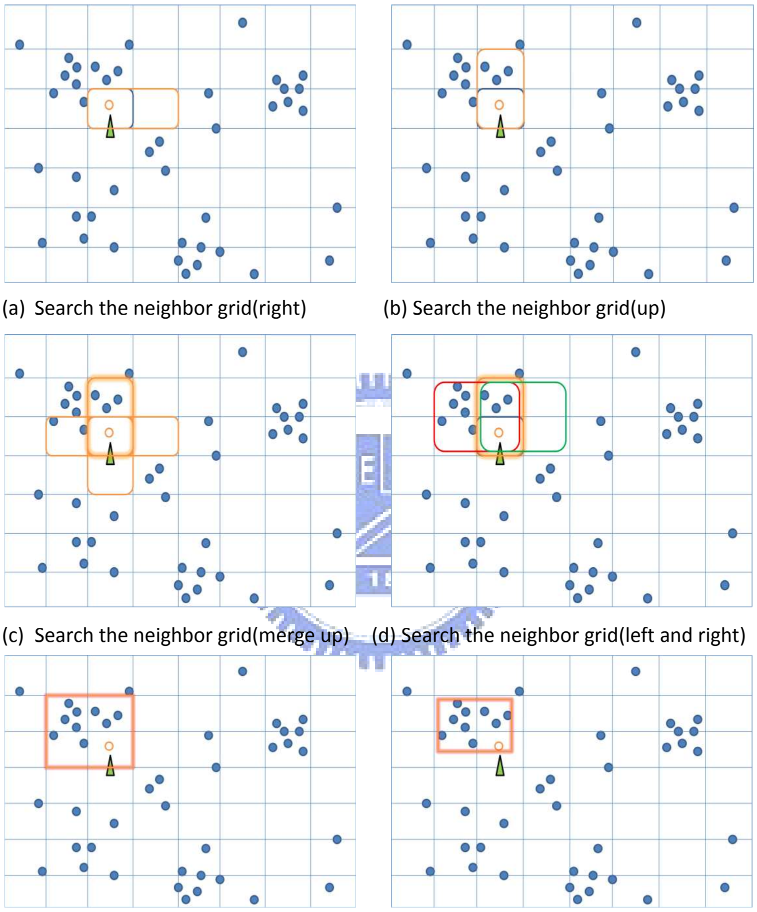
Figure 3.1: Example of cloaked region

scenario. After this phase was done, we can ensure that the MBR contains the most query points where property 1) is satisfied. We denote this MBR as M_1. Moreover, according to the distribution of the query points, some extreme conditions will occur. First, if too many query points gather at a small area, the cloaked region will be very small too, even if the value of k is large. It means that the cloaked region will be a very small area and the baleful entity can conjecture someone's location without knowing his/her identity. Hence Amin is introduced. With Amin, the cloaked region will have a lower bound. This will allow users can have basic location privacies when issuing LBS queries.

Algorithm 1 Region Generation

```
1: select the oldest query to serve as head, which has the $K$ requirement
2: search the users $u$ which has the same or smaller value of $k$ in the same grid
3: while $|u| \leq K/G$ do
4:     search neighbor grid and choose the grid that contains the most users
5:     if Property 1 can be satisfied then
6:         merge the grid and go to 4
7:     else
8:         choose another query point as the head, and go to 2
9:     end if
10: end while
```

Obviously, if the value of k or g is large, it is obviously that M_1 may not have enough users to satisfy the k-anonymity. Then we will go into the split-cloaking phase. In this phase, we



(a) Search the neighbor grid(right)

(b) Search the neighbor grid(up)

(c) Search the neighbor grid(merge up)

(d) Search the neighbor grid(left and right)

(e) Search finished

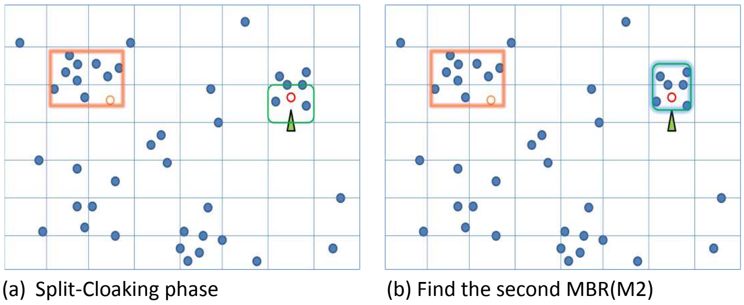
(f) Find MBR

Figure 3.2: Search phase

choose another query point which is not resided in M_1 to serve as the head, and then perform the search phase again. We choose the oldest query which is still not been cloaked because it will guarantee no query will be omitted. We perform search phase and split-cloaking phase repeatedly until all of these regions are found. We call these regions as candidate regions, denote as M_1, M_2,..., M_n. And then go into the second step: region selection step.

3.2 Region Selection: Optimization of the cloaked area and the relative k-anonymity

In Algorithm 1 we generate many cloaked regions which are called candidate regions. In the region selection step, our goal is to decrease the regions area, and meet the global disperse property. After region generation step finishes, we know that each region has its k value (k_i) and area (A_i). The challenge is how can we find the best regions that contain enough users and the total area is small. Remember that when the cloaked area is smaller, the return answer size will be smaller, too. In other words, this will improve the quality of the return answer size of the LBS query. So we need to pick up those regions which will have smaller area and still satisfy users' k-anonymity. In addition, we should distribute the cloaked regions (equal or more than g) in the whole region to meet the global disperse property. Figure 3.3 depicts this scenario.Here we can transform this problem into a knapsack problem, a traditional dynamic programming problem. Following comes the transformation process: User's k-anonymity (K) denotes as the capacity of the knapsack. Every separated region has its k value (k_i), i = 1...n. The value denotes as the article's volume. And each region's area (A_i) denotes as article's merit. $k_{1'} + ... + k_{n'} \geq K$, $A_{1'} + ... + A_{n'} = A_{total}$, $k_{1'}...k_{n'}$ are the best selection of $k_1...k_n$ such that A_{total} is minimum. Notice that in the knapsack problem, we need to pick up objects which total volume can't larger than the capacity of the knapsack. But in our scenario, we must pick the regions where the sum of the value $k_1 + ... + k_n$ should be equal or larger than K. Thus in the last we might need to select another region to fill up the difference of K - ($k_1 + ... + k_n$). We pick up the region that contains the oldest query to decrease eliminate the query starvation. Besides, the knapsack problem will conduct a maximum value of the total volume. But in our case the minimum is required instead. So we here we choose a value M_{max} from all the regions generated in the region generation step. The M_{max} value is the largest one among A_i. Finally each object's merit should be replaced as M_{max} - A_i. Through this transformation, we can derive an "near" optimal solution between the value of k and the


(a) Split-Cloaking phase

(b) Find the second MBR(M2)

Figure 3.3: Split-Cloaking phase

cloaked area.

3.3 Extension

In the above section, we can find out that if there are enough query points, the cloaking process can easily find other k-1 users to satisfy the k-anonymity constraint. But in fact, there are not always so many users issuing LBS queries. When the query points are not enough, the queries will fail even we apply the Split-Cloak algorithm. In this condition, we will adopt some other users as potential users. When there are not enough query points such that the cloaking algorithm can't contain enough other users, the anonimizer (the trusted entity) will broadcast a message to ask the potential users to report their locations. Remark that the property 1) can not be violated here. We can view the potential users as the same as the query users. There are just one difference between them: the return answers are invalid for potential users, since the potential users send no LBS queries indeed. Therefore, there will need no alteration in the region generation step, either the searching phase or the split cloaking phase. The anonymizer need to inform the potential users only. The process is quite the same as the search phase. In the first, when the anonymizer can not find enough cloaked regions to satisfy the user's k-anonymity, it will broadcast a message in the grid that the heads resides in. And all the potential users reside in this grid need to report their location to the anonymizer. Second, if the users are enough, the anonymizer will stop the broadcasting. If not, anonymizer will broadcast the message to the neighbor grids or choose another grid that another head resides in, and then choose the grid that contains the most users. This process is just the same as the split-cloaking phase does.

Here will exist a tradeoff that if all the points are query points, the return candidate answers will be all valid to all the users. But when we adopt the potential users who not issue

LBS queries to support cloaking process, the return answers will be invalid for them. The more potential users that we adopt, the more invalid answers will be generated. This means the LBS servers will consume more computing power on useless items. But this will lead no query failure indeed. And we provide benefits for the potential users. If the potential users report their position even not issue LBS queries, they will get higher priorities when they issue LBS services. This represents that when they really issue LBS services, our system will cloak their queries earlier.

Chapter 4

Performance Evaluation

4.1 Simulation Model

In this section, we consider a random walk model that is based on *random way-point* model in our experiments. At the beginning, the LBS clients are distributed in a spatial space of 300x300 randomly, and we divide the space into 20x20 uniform grids. Each mobile client chooses its destination in this space with a randomly determined speed from a uninform distribution $U(V_{min}, V_{max})$. When the client reaches its destination, it will pause for a time period to change the speed and choose its next destination. This time period is also determined from a uninform distribution $U(T_{min}, T_{max})$. The simulation parameters are shown in Table 4.1. We here consider that users all issue nearest-neighbor queries and they will issue queries after they reach their destinations. We will run the algorithm periodically or run it while the trusted anonymizer receives enough query users. Simulation will last for 600 seconds.

4.2 Evaluation Parameters

Here we address three measures of the proposed location k-anonymity model.

Tradeoff between privacy and quality: In Section 3.3, we adopt potential users to ensure the success of a query. But too many potential users involve in the cloaking process will cause low qualities of the return answers. Besides, the potential users need to reveal their location information to help another users who issue LBS services indeed.

Update message counts: The trusted anonymizer will be overloaded if there are too many update messages sent from the mobile users. The clients will consume more power when

	Default	
Nodes:	1000	500
Node speed:$[V_{min}, V_{max}]$	[5.0,10.0]	[15.0,30.0] m/s
Pause time:	Min time=0.0, Max time=10.0	
Simulation size:	300x300	
Simulation time:	600s	
Global disperse G:	3	
Amin:	25	

Table 4.1: Simulation Parameters

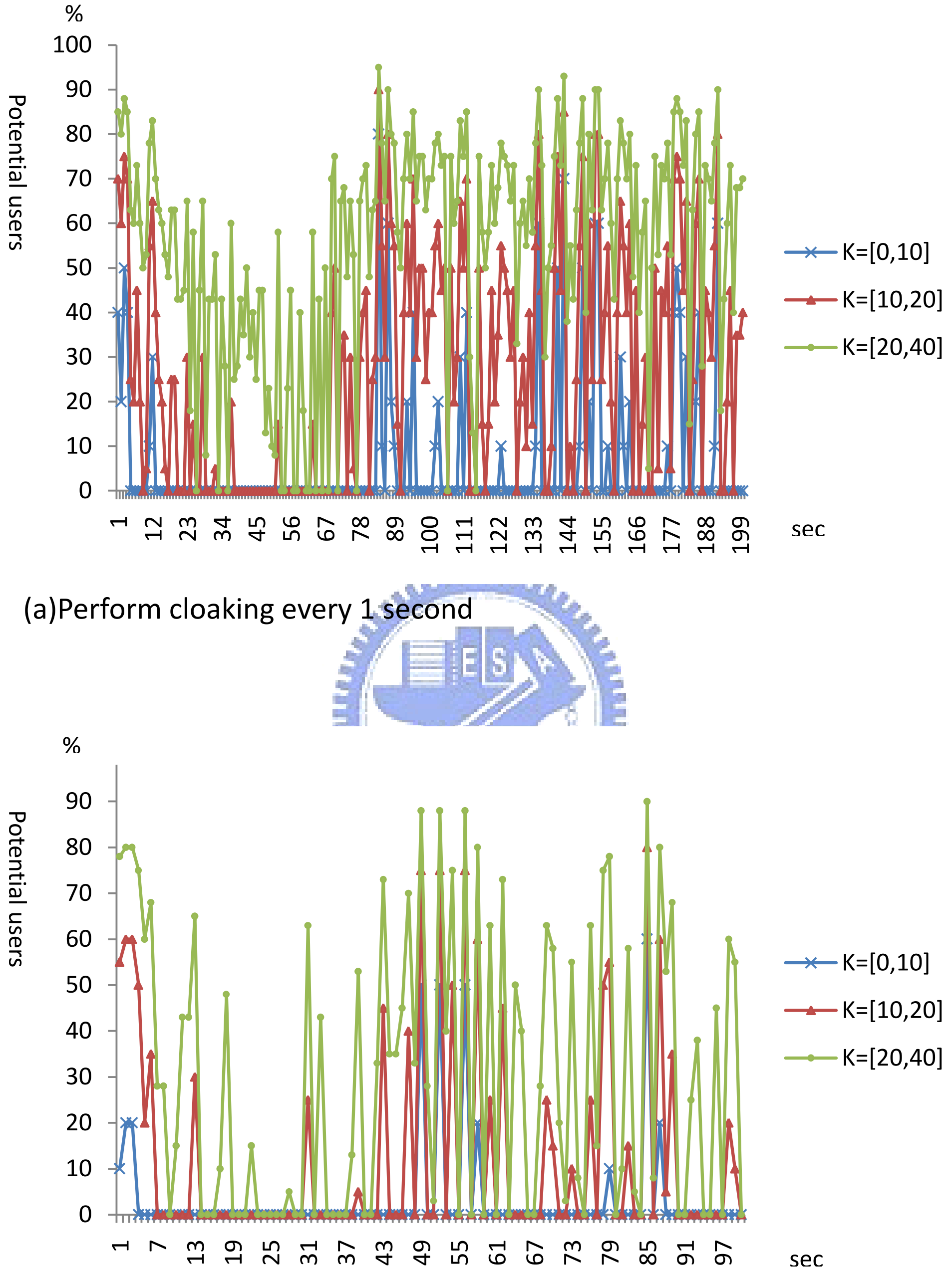
they send more update messages to the trusted anonymizer.

Spatial resolution: As mentioned above, we distribute the cloaked region into a set of smaller regions. If the sum of all these areas are smaller, we can conjecture that the return amount of candidates will be fewer.This is meaningful in both server side and client side. In server side, smaller area can save more computing power on searching candidate answers. So does the transmission cost that should be consumed on transmitting the answers back to the users. In client side, however, fewer candidate answers mean that the clients can filter out the useless information more effectively. This also can save energy for mobile devices. Besides, an additional goal is to reduce the sum of these regions' area.

In traditional 1-region search cloaking approaches, the most possible reason of a failure query is that they can not find enough other users in one cloaked region to satisfy the k-anonymity constraint. But in our approach, we expect that we can achieve relative high k-anonymity level because we divide the large K into several smaller ks and thus can easily find enough users nearby to perform the cloaking. Besides, we adopt potential users to ensure the success of queries even if the query points are few in entire area.

4.3 Tradeoff between privacy and quality

Figure 4.1 gives the percentage of the potential users involved in the cloaking with respect to varying the algorithm execution time period from 1s to 2s. With respect to the K-anonymity level, the usage of potential users increase as the value of K increase. It is easy to understand



(a)Perform cloaking every 1 second

(b) Perform cloaking every 2 seconds

Figure 4.1: K-anonymity level vs. Potential users

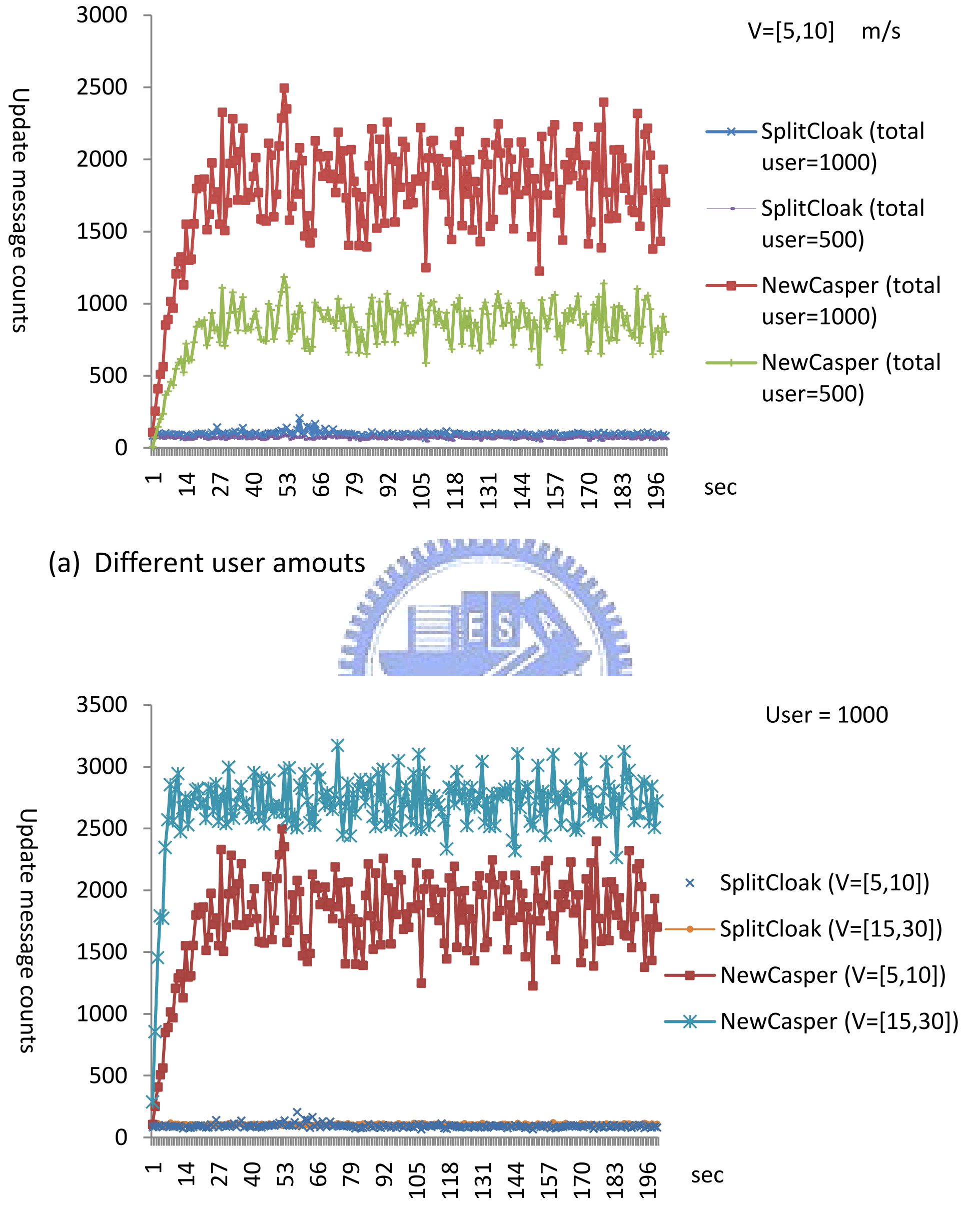
that if the users claim a high K-anonymity level, there will not have enough other query users to satisfy the K-anonymity constraint. Hence we will need more potential users to assist the cloaking. However, when we enlarge the algorithm execution time period from 1s to 2s, the percentage of potential users involved in the cloaking will decrease. This is because when we enlarge the execution time period, the anonymizer can receive more queries in the time period and thus will have more valid users to support the cloaking process. In addition, we here will find that if a user issues a LBS query with a large value of K (ex. 40), the query will still be satisfied while we should adopt more potential users.

4.4 Update Message Counts

Here we conduct a simulation about the update message counts between NewCasper and our work. In NewCasper, users must report their location information when they leave their current grids (location update). But in our work, users need to report their location information only when the trusted anonymizer broadcasts a require message to them. It happens when there are not enough other query users to perform the cloaking process. Figure 4.2(a) shows that location update will conduct a significant amount of update messages. Moreover, when the user amounts grow up, the update messages will also grow up significantly. But in our work, user reports its location information only when issue a LBS query or be asked by trusted anonymizer. Therefore, the user amount only makes little impact on the update message counts. Figure 4.2(b) describes the impact of the users' speeds. We can see that,in NewCasper, if users move with a relative high speeds, the update message will increase due to the location update. But in our scenario, users update their location only when they issue LBS queries. There is no relationship between the users' speeds and the update message counts. Thus there is no apparent variation in update message counts.

4.5 Spatial Resolution

Figure 4.3(a) shows that if the value of k is the same, larger period of time to perform the cloaking will conduct smaller cloaked area. It is because when we perform the cloaking algorithm with a large period of time will let the trusted anonymizer receive more LBS queries. This will reduce the need of potential users. And because each region satisfies the *Local Dense* property, it conduct smaller region area than the 1-region search approaches. We easily can



(a) Different user amouts

(b) Different [V*min*,V*max*]

Figure 4.2: Update massage counts

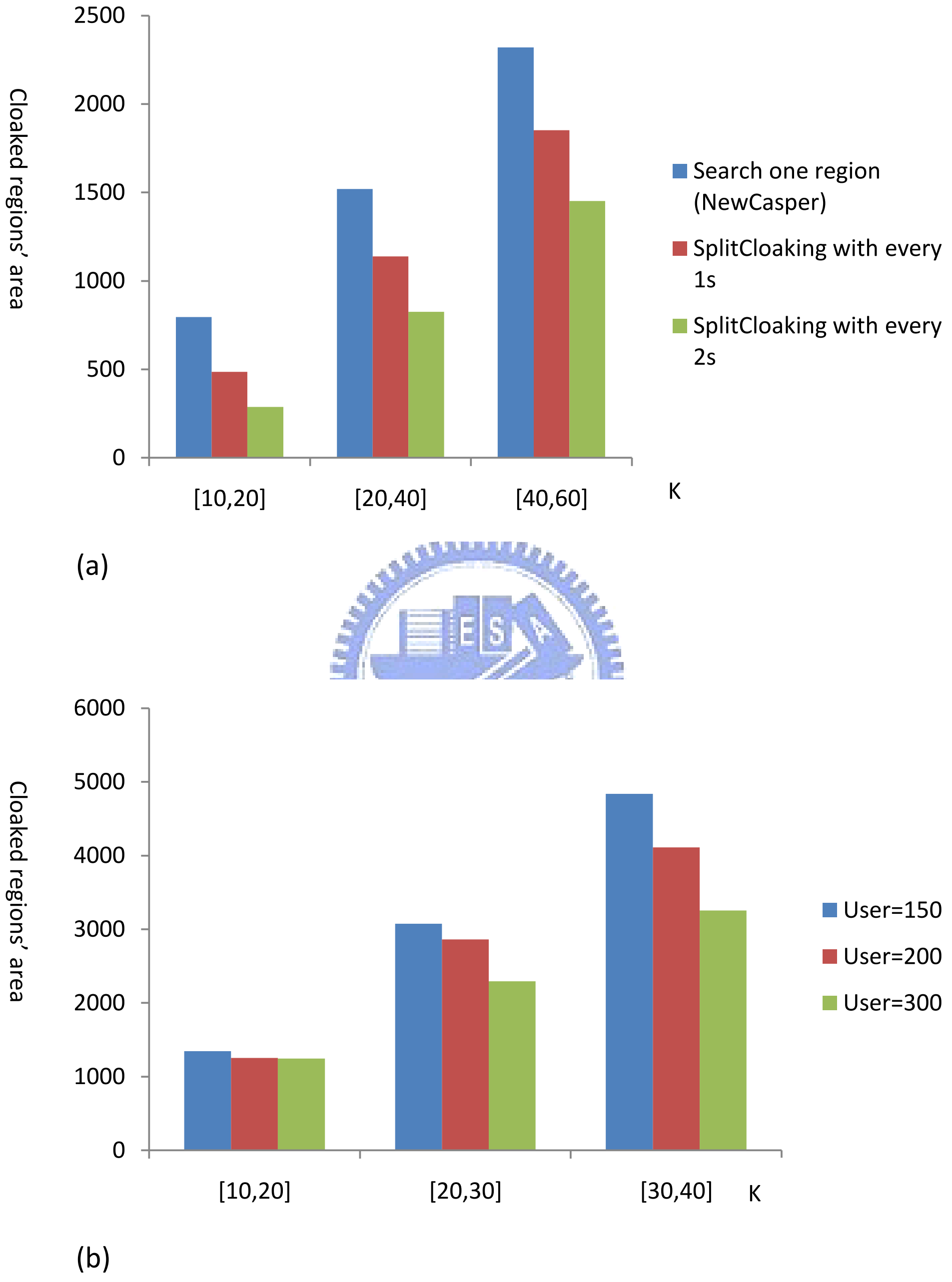


Figure 4.3: Spatial resolution

find that when the value of k gets larger, the cloaked area increase properly. However, there still exists a meaningful result concealed in this graph. That is even the value of k reaches a great number, the split cloaking process still can find enough cloaked region to support users' k-anonymity. In other words, the split cloaking algorithm will work even if the users select a large value of k. Nevertheless, the algorithm will contain more cloaked regions rather than let the query fail. In Figure 4.3(b), we delay the cloaking process until the trusted anonymizer receives $< 150, 200, 300 >$ LBS queries. We can find that higher query density will conduct smaller cloaked area, and smaller cloaked area will conduct a smaller return answer size from LBS providers. Another benefit from delaying the cloaking is that we use almost no potential user in the cloaking process. This means the return candidate answers are all valid here.

Chapter 5

Conclusion

We propose a new concept for protecting location information by means of split cloaking. With this approach, we can easily achieve the k-anonymity even if the value of k is very large. We also introduce the potential users to assist the cloaking process to ensure the success of LBS queries. Besides, we avoid the defects of using dummies which will lead to an inefficient system. Moreover, we just need to make little alteration on clients and LBS servers and this is a very important feature in system implementation.

Bibliography

[1] Bhuvan Bamba, Ling Liu, Peter Pesti, and Ting Wang. Supporting Anonymous Location Queries in Mobile Environments with PrivacyGrid. In *International World Wide Web Conference (WWW)*, 2008.

[2] Chi-Yin Chow and Mohamed F. Mokbel. Enabling Private Continuous Queries For Revealed User Locations. In *International Symposium on Spatial and Temporal Databases (SSTD)*, 2007.

[3] Chi-Yin Chow, Mohamed F. Mokbel, and Xuan Liu. A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services. In *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, (ACM-GIS)*, 2006.

[4] Bugra Gedik and Ling Liu. A Customizable k-Anonymity Model for Protecting Privacy. In *International Conference on Distributed Computing Systems (IEEE ICDCS)*, 2005.

[5] Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. PRIVE: Anonymous Location-Based Queries in Distributed Mobile Systems. In *Proceedings of International World Wide Web Conference (WWW)*, 2007.

[6] Ali Khoshgozaran and Cyrus Shahabi. Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy. In *International Symposium on Spatial and Temporal Databases (SSTD)*, 2007.

[7] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. Anonymous Communication Technique using Dummies for Location-based Services. In *IEEE International Conference on Pervasive Service (ICPS)*, 2005.

[8] Wei-Shinn Ku, Roger Zimmermann, Wen-Chih Peng, and Sushama Shroff. Privacy Protected Query Processing on Spatial Networks. In *International Workshop on Privacy Data Management (PDM)*, 2007.

[9] Mohamed F. Mokbel. Towards Privacy-Aware Location-Based Database Servers. In *Proc. of the Second International Workshop on Privacy Data Management (PDM)*, 2006.

[10] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. The New Casper: Query processing for Location Services without Compromising Privacy. In *International Conference on Very Large Data Bases (VLDB)*, 2006.

[11] James L. Sweeney. K-anonymity: A model for protecting privacy. In *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS)*, 2002.

[12] Zhen Xiao, Xiaofeng Meng, and Jianliang Xu. Quality Aware Privacy Protection for Location-based Services. In *International Conference on Database Systems for Advanced Applications (DASFAA)*, 2007.