Privacy Protected Location based Services

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Jie Bao

Certificate of Approval:

Qin Xiao
Assistant Professor
Department of Computer Science and Software Engineering
Auburn University

Wei-Shinn Ku, Chair
Assistant Professor
Department of Computer Science and Software Engineering
Auburn University

Cheryl D. Seals
Assistant Professor
Department of Computer Science and Software Engineering
Auburn University

George T. Flowers
Dean
Graduate School
Auburn University

Privacy Protected Location based Services

Jie Bao

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
December 18, 2009

Privacy Protected Location based Services

Jie Bao

Signature of Author

Date of Graduation

Vita

Jie Bao, son of Zheming Ni and Jianqiang Bao, was born May 26, 1985, in Hangzhou, Zhejiang, China. He earned his Bachelor's degree in Computer Science and Software Engineering from Zhejiang University, Zhejiang, China in 2007. Before his graduation in Zhejiang University, he studied in Singapore Management University as an exchange student for one year.

Thesis Abstract

Privacy Protected Location based Services

Jie Bao

Master of Science, December 18, 2009
(B.S., Zhejiang University, 2007)

72 Typed Pages

Directed by Dr. Wei-Shinn Ku

An increasing number of the mobile devices nowadays embedded with the GPS module (e.g., smart phones, PDAs and RFIDs), which makes the user can facilitated from the location based services. They can ask for the nearby points of interest (POIs) which can be gas stations, restaurants and track the trace of the buses. Example of such services likes "show me the nearest gas station".

In order to access location-based services, mobile users have to disclose their exact location information to service providers. However, adversaries could collect the location information for purposes against mobile users' privacy, such as tracking and stalking.

The most popular solutions for privacy protection are utilizing the K-anonymity model to blur user's exact location information. By using this principle, the client will not send its exact location information to the service provider, but a blurred region with at least $k-1$ other peers. As the result, the services provider will not be able to find out the identity of the query issuer, even if they know the exact user distribution in that area.

There are two very popular system architectures applying the K-anonymity principle to construct the cloaking region for the privacy preserving spatial queries: the centralized spatial cloaking and the peer to peer spatial cloaking. However, there are some drawbacks and defects for these existing solutions. For example the central server for the first solution

will become a single point of failure and performance bottleneck. And for the Peer to Peer solution, there are several other privacy issues such as the distinguishability for the peers.

This research work proposes a cache management mechanism for the centralized solution to further improve user privacy protection, saving computational power, and decreasing communication costs.

And for the decentralized solution, we propose a CAN (Content Addressable Network) based road network partition and an incremental query processing mechanism to extend the location privacy protection over the road networks and improve the indistinguishability for K-anonymity principle. And a corresponding k nearest neighbors searching algorithm is also proposed to optimize the existing solution.

Acknowledgments

I would like to express my deep appreciation to Dr.Wei-Shinn Ku for the guidance and support he has provided throughout my graduate study at Auburn University.

I would also like to express my gratitude to the advisory committee members, Dr. Xiao Qin and Dr.Cheryl D. Seals.

Above all, I would like to thank my parents who helped me come to U.S. and pursue graduate study in the Computer Science and Software Engineering Department at Auburn University.

Style manual or journal used Journal of Approximation Theory (together with the style known as "aums"). Bibliograpy follows van Leunen's *A Handbook for Scholars.*

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `aums.sty`. The images and plots were generated using Microsoft®Visio 2003 and Microsoft®Excel 2007.

Table of Contents

List of Figures

Chapter 1

Introduction

As the result of recent advances in wireless technologies, more and more personal mobile devices (e.g., cell phones, PDAs, etc.) possess the ability to access the Internet ubiquitously. In addition, Global Positioning System (GPS) receiver modules are becoming a standard component in the recent generation hand-held devices. Consequently, novel location based services (LBS) allow users to launch location-dependent queries ubiquitously. Sample queries of such location based services include find me the nearest ATM and show me the gas station with the lowest price within one mile. In order to fulfill these queries, mobile users have to reveal their current location information to service providers. However, service providers may disclose the trajectory of a certain user to malicious users which can decrease the dependability of LBS.

For protecting mobile users privacy, recent research in [12] proposed a framework for location based services without compromising location privacy by leveraging the K-anonymity concept [15].

There are two typical solutions to implement the K-anonymity mechanism, the centralized spatial cloaking and the decentralized solution.

For the centralized framework, it contains a trusted server to collect user location information and perform cloaking procedures. Then, the trusted server will send the location dependent query along with the cloaked spatial area to service providers to retrieve query results. The returned query solutions will be sent back to individual users by the trusted server as well. The problem for this kind of solution is that the location anonymizer is always considered to be the performance bottleneck, because all the queries will go through it. And for each query, the location anonymizer have to run the cloaking algorithm and construct the cloaked region individually.

Since the trusted server has the knowledge of all query results, we propose to store them in memory and use the cached data to answer future queries. Our solution has two main advantages. First, user privacy protection can be further improved, because the trusted server does not have to forward every query to service providers and it is much more difficult for adversaries to launch correlation attacks [3]. Second, with our cache management techniques, fewer queries have to be answered by service providers. Consequently, computational resources and communication costs can be effectively saved.

For the decentralized solution, the spatial cloaking process depends on the collaboration of the peers. The peers in the frame are able to talk to each other, and share their location information. It doesn't need the trusted third party as the middle man. And as the result, it avoids the single point of failure and the performance bottleneck. However, the previous peer to peer framework [4] also has some drawbacks. First of all, it organizes the peers in the Euclidean space and constructs the cloaking region using grids. In this research work, we further extend the peer to peer framework to road networks. The CAN (Content Address Network) [26] mechanism is applied to divide the road networks and organize peers. In that way, we are able to further improve the indistinguishability for the K-anonymity principals that we can exclude some peers who are not qualified for the K-anonymity. And the corresponding spatial query processing will be done in a incremental way to blur the user's exact location. Because we used road networks to organize the peers, we are able to get the nearest neighbor result on the road networks which makes the result more applicable.

The rest of the thesis is organized as follows: Chapter 2 introduces the basic background information about location based services and the potential threats. Chapter 3 surveys the related work of privacy protected query processing and describes two of the most popular solutions of spatial cloaking. The Cache based technology is detailed in Chapter 4 including its experiment results. Road network based solution is presented in Chapter 5. We conclude this research in Chapter 6, and we also raise some open issues for future research. And all the references are listed in the bibliography.

Chapter 2

Motivations

2.1 Location Based Services

Definition 1:

A location-based service (LBS) is an information and entertainment service, accessible with mobile devices through the wireless network and utilizing the ability to make use of the geographical position of the mobile device [19].

Definition 2:

A wireless-IP service that uses geographic information to serve a mobile user. Any application service that exploits the position of a mobile terminal.

These definitions are quite similar, that the location based service is a intersect of Web/Data services, GIS services, Mobile and Wireless communication services as shown on Figure 2.1.

It is because, historically speaking, location based services were applied commercially firstly in Japan by DoCoMo which is a mobile carrier company all the way back to July 2001. And the first mobile phones equipped with GPS is announced by KDDI in December 2001 [18]. Mobile device venders have tended to take "upstream initiative", which is try to embed GPS modules in their mobile equipment. So, originally, LBS was developed by mobile carriers in partnership with mobile content providers.

As mentioned before, the location based service is the mobile web services, essentially, with the extra help from the poisoning devices. So, as the consequences, the typical location based services is very similar to these mobile web/data services which are consisted with four parts:

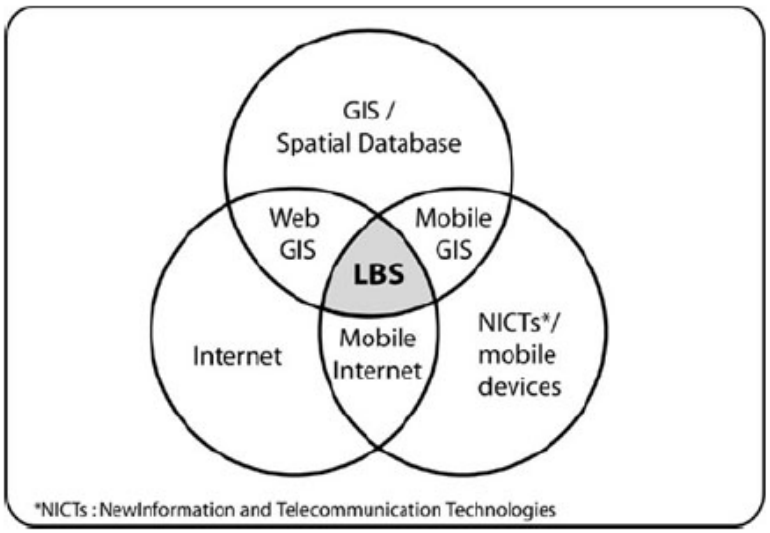


Figure 2.1: Location based Service is a intersect role.[19]

1. Mobile Users
2. Positioning Module
3. Wireless Communications
4. Service and Date Providers

The main advantage is that mobile users don't have to manually specify ZIP codes or other location identifiers to use LBS, when they roam into a different location. And in many cases, the users are not able to tell where the exact location they are. GPS tracking is a major ingredient for making it possible, utilizing access to mobile web.

LBS services include services to identify a location of a person or object, such as discovering the nearest banking cash machine or the whereabouts of a friend or employee. LBS services include parcel tracking and vehicle tracking services. LBS can include mobile commerce when taking the form of coupons or advertising directed at customers based on

their current location. They include personalized weather services and even location-based games [16] . And there are several different examples of Location Based services:

Resource tracking with dynamic distribution. Taxis, service people, rental equipment, doctors, fleet scheduling.

Resource tracking. Objects without privacy controls, using passive sensors or RF tags, such as packages and train boxcars.

Finding someone or something. Person by skill (doctor), business directory, navigation, weather, traffic, room schedules, stolen phone, emergency calls.

Proximity-based notification (push or pull). Targeted advertising, buddy list, common profile matching (dating), automatic airport check-in.

Proximity-based actuation (push or pull). Payment based upon proximity (EZ pass, toll watch).

2.2 Privacy Concerns in Location Based Services

Along with all the benefits and convenience the location based service have provided, there are many privacy issues and concerns regarding this kind of technology.

Information that users used in the location based services is considered as a very personal and critical information for many reasons. Users' information included in the location based services consisted with four major components:

- Users' identity information.
- Users' location information - the location coordinates.
- The context of the location based services.
- Timestamp.

Some combinations of the information may tell the malicious user some private knowledge. For example, the location information and the timestamp will draw your travel trace or even the life pattern. The context of the location query may leak some critical information like medical conditions.

This is not a imaginary vulnerability nowadays. But as a matter of fact, there are a lot of real crime examples.

For example, an adversary may check a user's habit and interest by knowing the places she seeks. In many real life scenarios, GPS devices have been used in stalking personal locations [27], [28]. According to the recent reports[29], stalking and harassment taking advantage of GPS devices and location related services is becoming very serious and significant. "Domestic violence is responsible for 2 million injuries and nearly 1300 deaths among U.S. women age 18 and older, according to a division of the U.S. Centers for Disease Control and Prevention"[29].

And stalking and tracing is only one quite oblivious security problem. Triangulation and other techniques can be used to increase the accuracy of location; and some third-generation mobile phones have an integrated GPS receiver which provides location information with an accuracy of a few meters.

Most of the data service providers will make log files to keep track of the system and the queries, And users' critical information will be protected. However, with some untrustable service providers, malicious users may compromise the server and be able to decrypt and access users' sensitive information about individuals based on their issued location-based queries. In this way, the malicious user will know your locations and

The information leakage during the query transmission is another potential privacy issue. As mentioned before, the content of the location dependent queries is also a critical information. Due to the natural of the wireless communication, malicious user can always overhear these location based queries if they are close enough. The content of the location

based queries might leak during the information transfer. If you're querying for the availability if some kind of medicine, it may release your medical conditions to the malicious users.

So, in order for the users to take the advantage of this technology, privacy protection is very necessary and urgent.

Chapter 3

Related Work

As we presented in the previous chapter, privacy protection is very necessary and urgent in the location based services.

And a lot of related research and projects have been done. However, the conventional privacy protection mechanisms highly rely on encryption and pseudonym techniques to safeguard users' communication and hide users' identities. However, the queries launched by users may contain other sensitive information (e.g., physical locations), which potentially could hurt users' privacy. Recently, several novel techniques [8], [6], [12], [13] have been proposed to protect users' privacy for location-based spatial queries without compromising privacy based on the well-known K-anonymity mechanism [15]. A user can request for a cloaked area which cover the location of at least K - 1 closest peers to blur its actual location. In order to keep a reasonable size of the cloaked region in the high user density areas, the user is able to decide the minimum acceptable cloaked region size. Their system model is similar to the architecture depicted in Figure 3.1. However, each of them applies different cloaking mechanisms, algorithms and user location management data structures. In order to avoid the single point of failure (the location anonymizer) problem in the aforementioned systems, Ghinita et al. [6] and Chow et al. [4] proposed peer-to-peer architecture based spatial cloaking techniques. On the location-based service provider side, there these system ([8], [6], [12], [13],[20]) also provide solutions for cloaked nearest neighbor queries and range queries. For cloaked range queries, the general solution is to extend the received cloaked region outward by the search distance d on all dimensions. For cloaked nearest neighbor queries, most existing solutions are based on the range nearest neighbor technique [5], which retrieves the nearest neighbors for every point within a range.

3.1 Centralized Spatial Cloaking

The centralized spatial cloaking mechanism is using a trusted third party as the middle man between users and the service provider which is always referred as the location anonymizer or cloaker([6], [12], [20], [21]). It will get the real query information from the user and blur it based on user's customized privacy profile which usually include the minimal K value and minimal size of the cloaked region. After the spatial cloaking process, the location anonymizer will send the modified query to the service provider and get the result back to the user.

3.1.1 Centralized System Architecture

We'll present a very classic centralized spatial cloaking solution, Casper, which is proposed by Mokbel, Chow and Aref [12] as the example. In the Casper design, every user will maintain its own privacy profile which specifies the minimal k value for K-Anonymity and a minimal cloak region size A_{min}. And user report its location information to the centralized location anonymizer periodically.

Figure 3.1 demonstrates the basic system architecture for the centralized spatial cloaking mechanism. It consists two main components: the location anonymizer and the privacy-aware query processor. The location anonymizer maintain users' location information and it get the query requests. It anonymizes the location to a relatively bigger region which satisfies user's privacy profile (both k and A_{min}). k indicates that the user wants to be k-anonymous, that is, indistinguishable among k users, while A_{min} specifies the minimum resolution of the cloaked spatial region. The larger the value of k and A_{min}, the more strict privacy requirements a user needs. And users have the ability to change their privacy profile at any time. After the cloaking region is successfully constructed, it will send the query to the service provider using the blurred region.

The other component, the privacy-aware query processor is embedded in the service provider to process the location based query anonymously. It is specially designed, because

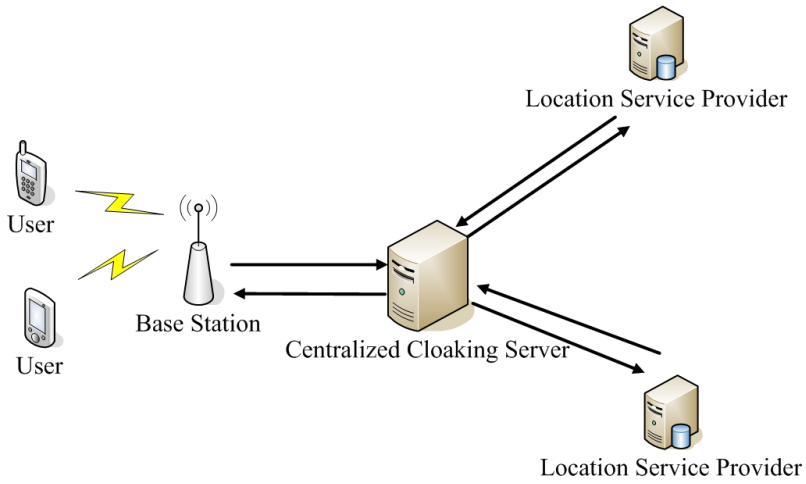


Figure 3.1: Centralized Spatial Cloaking System Architecture.

the privacy-aware query processor deals with the blurred query region rather than the exact location point. Instead of return the exact answer, the processor will return a list of candidate points to answer user's query though the location anonymizer. But it will guarantee that the exact answer will be included in the candidate list. After the user gets the candidates' list, the user will figure out the exact answer himself with his own exact location information.

3.1.2 Location Anonymizer

Figure 3.2 illustrates the data structure for the location anonymizer. The main idea is applying a grid-based complete pyramid architecture that hierarchically decomposes the Euclidian space into H levels where a level of hight h has 4^h grid cells. And the relationship of the cells in different levels is shown in Figure 3.3 that cell 0, 1, 2, and 3 are the lower level cells and they four together will be an upper level cell. And the most up level of the whole structure is the root of the pyramid which will cover the whole space. Each pyramid cell

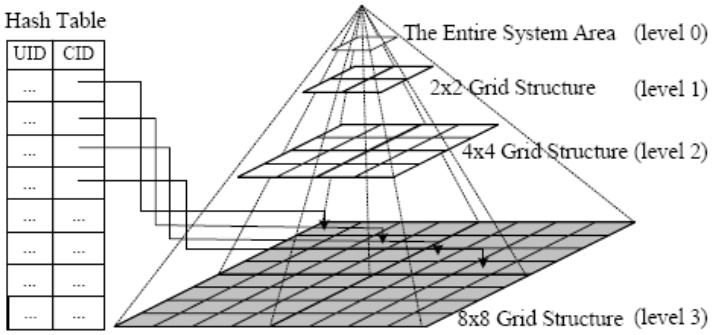


Figure 3.2: Pyramid Indexing Structure. [12]

will be represented as (*cid, N*) where *cid* is the identifier and N is the number of mobile users inside area that cell covers. The location anonymizer will keep track and update the current user numbers insider each cell.

Adaptive Location Cloaking Data Structure

Figure 3.3 depicts the data structure for the adaptive spatial cloaking which mainly uses the previous pyramid structure. And it is an updated version of the pyramid structure. The content is exactly the same as those in Figure 3.2. The main idea of the incomplete pyramid structure is to maintain only those cells which will be potentially used as the cloaking region for the users. For example, if the user have a restrict privacy profile that the cells in the lowest level will not satisfy any cloaking request, the location anonymizer will not maintain the information in that lowest level. In that way, the location anonymizer will save the computational and communicational overhead for location update as well as the cloaking requests.

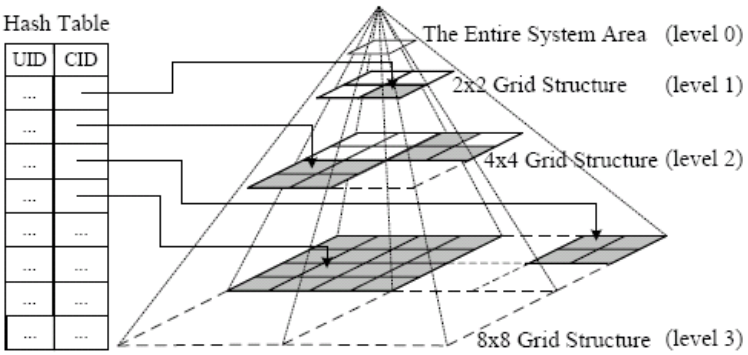


Figure 3.3: The adaptive location anonymizer.[12]

Maintenance

Because the location anonymizer will keep track of the current user number in each grid cell, a large amount of update communication is expected. In order to reduce the communication overhead, a location update is sent to the location anonymizer will in the form like *(uid, x, y)*, where *uid* is the identifer of the user while x and y are the spatial coordinates of the user's new location. Whenever, the location anonymizer gets the update request, it will first launch a hash function *h(x, y)* to get its cid_{new} at the lowest level of the pyramid hierarchy. Then, the location anonymizer will retrieve the user's entry in its database and get the original cell identifier cid_{old}. If the old cell identifer is the same as the new one $(cid_{new}=cid_{old})$, which means the user is still in the old grid cell and there will be no more processing for the location anonymizer at this point. If a change is made $(cid_{new} \neq cid_{old})$, three operations will take place in the location anonymizer.

1. Update the new cell identifer in the hash table.

2. Update the counter N in both old and new grid cells.

3. If necessary propagate the changes in the cell counters N for the higher level pyramid grid cells.

If a new user is registered, a new entry will be created in the hash table and the counters of all the affected cells in the hierarchy will be increased by one. Similarly, when a existing user is going off-line or out of service boundary, the entry in the hash table will be revoked and the affected cells will be decreased by one.

And there are two advance operation called *Cell Splitting* and *Cell Merging* which are dedicated for the adaptive version of the pyramid structure.

Cell Splitting: A cell *cid* at level i needs to be split into four cells at level $i + 1$ if there is at least one user u in *cid* with a privacy profile that can be satisfied by some cell at level $i + 1$.

Cell Merging: Four cells at level i are merged into one cell at a higher level i - 1 only if all the users in the level i cells have strict privacy requirements that cannot be satisfied within level i.

The Cloaking Algorithm

Algorithm 1 described a bottom-up cloaking algorithm for a grid-based pyramid structure. The input for the function is the user's privacy profile (k, A_{min}) and the cell identifer *cid* for the location anonymizer to locate where the user currently is. After the location anonymizer gets the parameters, if the cell already satisfy the user's privacy profile, i.e., $cid.N \geq k$ (the user number in current cell is greater than k) and $cid.Area \geq A_{min}$ (the spatial size of the current cell is larger than the A_{min}), the location anonymizer will return the current cell as the cloaked region. If it is not the situation, we'll check its horizontal or vertical neighbor first. As it shown in Figure 3.4, horizontal and vertical neighbor of the cell is considered as the cells which belong to the same upper level cell and in the same

Algorithm 1 Bottom-up cloaking algorithm

```
1: FUNCTION Bottom-Up Cloaking (k,A_min, cid)
2: if cid.N ≥ k and cid.Area ≥ A_min then
3:    return Area(cid)
4: end if
5: cid_V ← vertical neighbor cell of cid
6: cid_H ← horizontal neighbor cell of cid
7: N_V = cid.N+cid_V.N, N_H = cid.N+cid_H.N
8: if (N_V ≥ k OR N_H ≥ k) AND 2cid.Area ≥ A_min then
9:    if (N_H ≥ k AND N_V ≥ k AND N_H ≤ N_V) OR N_V <k then
10:      return Area(cid) ∪ Area(cid_H);
11:   else
12:      return Area(cid) ∪ Area(cid_V);
13:   end if
14: else
15:   Bottom-Up Cloaking((k, A_min), PARENT(cid));
16: end if
```

row (horizontal neighbor) or column (vertical neighbor). For example the input *cid* represents cell 0, cell 1 will be its vertical neighbor and cell 2 will be its horizontal neighbor. If any combination of the horizontal or vertical neighbor will satisfy user's privacy profile, the anonymizer will return it as the cloaked region. If none of the combination will work, the algorithm will recursively execute with the parent cell of *cid* until the user's privacy requirements (both k and A_{min}) are fulfilled.

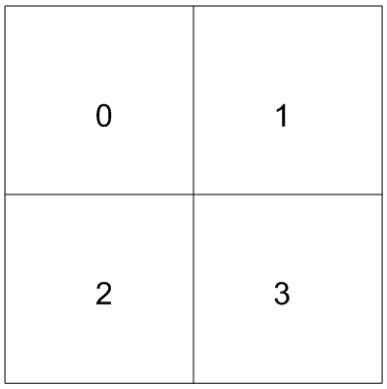


Figure 3.4: Examples of horizontal and vertical neighbores.

3.1.3 Privacy-Aware Query Processing

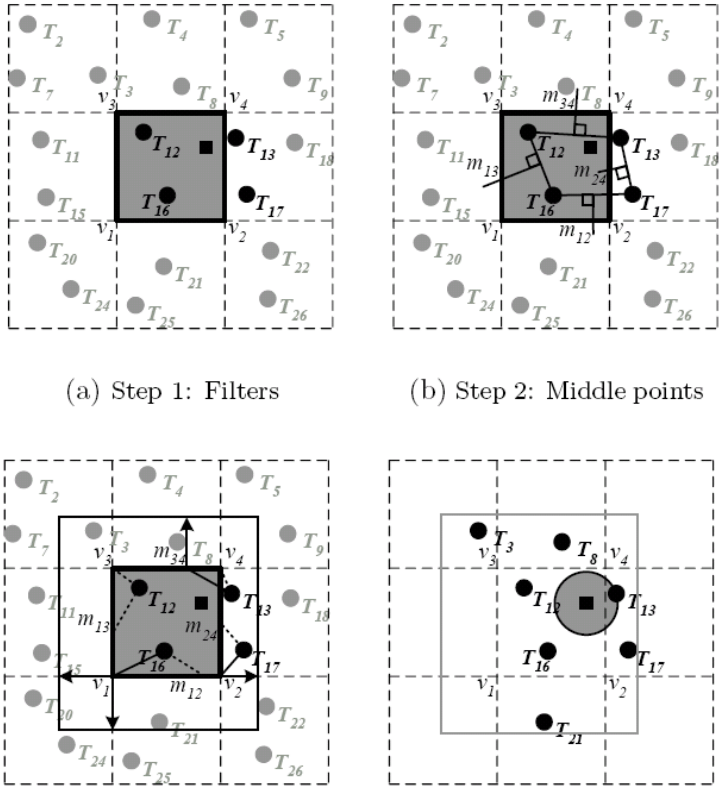
As we mentioned in the previous section, a privacy-aware query processor is needed to process the location-based query with the blurred region. The main goal of the processor is to provide highly efficient, accurate and anonymous location-based services. Because the query will not contain the exact location coordinates for the user but a cloaked region, the processor will not able to provide the exact answer but a list of candidate points.

The most common kind of query in the scenario is the private queries over the public data. User's location information can't be revealed and is considered to be confidential, while the location information for the POIs (Points of Interests) are widely available and is considered as public. Example of this kind of queries is like a person (private entity) asking for his nearest gas station (public entity). In this scenario, the location of the gas stations are known but the location of the query issuer is blurred to be a region.

As illustrated in Figure 3.5, there are 4 steps to find out the candidates for a private query.

The filter step. In this step, four filter target objects are chosen as the nearest point t_i for each vertex v_i in the cloaked region A. In the example, (Figure 3.5a), the four filters are T_{16}, T_{17}, T_{12}, and T_{13} where they are the nearest candidate points to the vertices v_1, v_2, v_3, and v_4, respectively.

The middle point step. In this step, we will try to find the point m_{ij} for each edge $e_{ij} = v_iv_j$ such that m_{ij} divides e_{ij} into two segments v_im_{ij} and $m_{ij}v_j$. The basic idea is that any point in the first segment v_im_{ij} will have t_i as its nearest filter target object and any point in the second segment $m_{ij}v_j$ will have t_j as its nearest filter target object while point m_{ij} is of equal distance from both targets t_i and t_j. We distinguish between two cases based on whether the two vertices v_i and v_j have the same filter or not. If v_i and v_j have the same filter t, then point m_{ij} does not exist as all points on edge e_{ij} will have t as their nearest target object. If t_i and t_j are different, m_{ij} is found by connecting t_i and t_j through a line L_{ij}. Then, another line P_{ij} is plotted



(a) Step 1: Filters

(b) Step 2: Middle points

(c) Step 3: A_{EXT}

(d) Step 4: Client

Figure 3.5: Example of a private query over public data.[12]

that is perpendicular to L_{ij} and divides L_{ij} into two equal segments. Finally, m_{ij} will be the intersection point between P_{ij} and the edge e_{ij}. Figure 3.5b depicts this step in our running example where points m_{12}, m_{13}, m_{24}, and m_{34} are plotted.

The extended area step. In this step, for each edge e_{ij}, we will try to find out the largest distance max_d from any point on e_{ij} to its nearest filter target object. Only three points on e_{ij} can be candidates to have the distance max_d to their nearest filter object, v_i, v_j, or m_{ij} . Thus, we compute the three distances d_i, d_j , and d_m that represent the distances from points v_i, v_j, and m_{ij} to targets ti, t_j , and t_i, respectively. Notice that in case m_{ij} does not exist, the distance d_m will be 0. Then, the distance max_d is computed as the maximum distance of d_i, d_j, and d_m. Finally, the area A_{EXT} is expanded by the distance max_d in the same direction of edge e_{ij}. Figure 3.5c depicts this step where all the computed distances for the four edges are plotted. Only those distances that contribute to max_d are plotted as solid while other distances are plotted as dotted lines. An arrowed line is plotted from each edge to represent its max_d extension to plot A_{EXT}. The intersection of the arrowed line with its edge represents the point that has contributed to max_d.

The candidate list step. In this step, the server issues a range query that returns all target objects within the area A_{EXT} as the candidate list. The candidate list is sent to the client where the query can be evaluated locally. Figure 3.5d depicts this step where the candidate list has only seven objects that include the exact query answer T13. Notice the difference between Figure 3.5d and Figure 3.5c, where in the former, the client needs to evaluate her query on only 7 targets as opposed to 32 targets in the latter case.

3.2 Peer to Peer Spatial Cloaking

Unlike the centralized spatial cloaking mechanism, the peer to peer spatial cloaking system doesn't have the trusted third part to construct the cloaked region. Because the

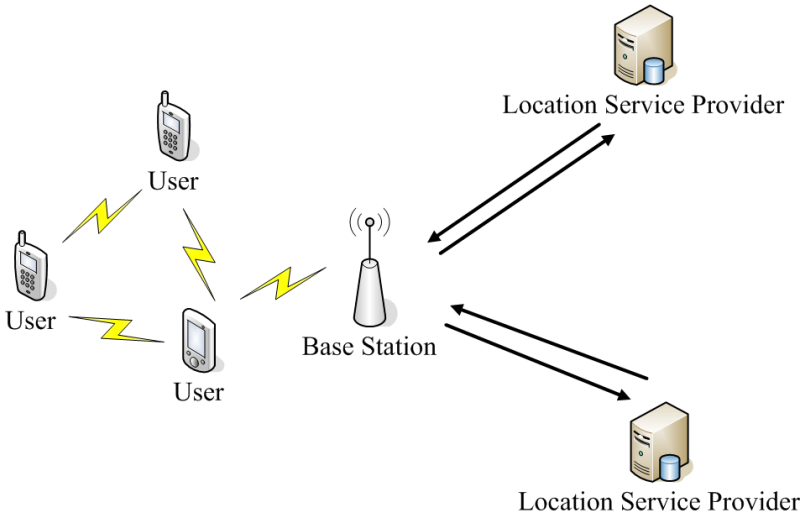


Figure 3.6: Peer to Peer Spatial Cloaking System Architecture.[4]

trusted third party, the location anonymizer, is not only a performance bottleneck but also a oblivious attacking target for the malicious user, the peer to peer spatial cloaking tries to construct the cloaked region by peer themselves. There are several solutions proposed using the peer to peer spatial cloaking idea ([4], [7], [22], [23], [24]).

In remain content of this section, a very typical peer to peer spatial cloaking system which is proposed by Chow, Mokbel and Liu [4] will be presented in details.

3.2.1 Peer to Peer System Architecture

The peer to peer system architecture is illustrated in Figure 3.6. Comparing with the centralized spatial cloaking (Figure 3.1), the peer to peer system doesn't have the location anonymizer as the middle-ware between the base station and the service provider. The cloaked region is constructed by users' collaboration and is sent to the service provider directly.

Consequently, the peer to peer spatial cloaking system contains two main components: *mobile clients* and *location-based database server*. Every peer in the system maintains its own privacy profile including two parameters , k and A_{min} like the centralized solution.

Basically, the system assumes that users in the system are able to communicate with each other via the wireless connections like bluetooth or IEEE 802.11 protocols. Each user also have the ability to communicate with the service provider directly and anonymously. And the client device will embedded with a positioning module (e.g., GPS module) to get the location information. The cloaked region will be constructed with the collaboration of the peers.

After the cloaked region is constructed the user will send the modified query to the base station anonymously. The base station will forward the query to the service provider. Similar to the centralized solution, the service provider should embedded with the privacy-aware processor with the ability to deal with the location based query with the blurred location information.

3.2.2 Peer to Peer Cloaked Region Construction

Data Structure

The entire area that the peer to peer spatial cloaking system covers is divided into grid. Users in the peer to peer system communicate with each other to discover other *k-1* peers to meet the k-anonymity requirement described in their privacy profile. In that way, user can hid his exact location into a cloaked spatial region that is the minimum grid area covering at least the *k-1* peers and itself, and satisfies the minimal area requirement(A_{min} in the privacy profile) as well. Moreover, each user maintains another parameter h that is the required hop distance of the last peer searching. The initial value of h is assumed to be one, which means the group forming function starts from the minimal circle.

Cloaking Algorithm

Figure 3.7 gives a running example of the peer to peer cloaking algorithm. In the scenario, there are 15 mobile users in the system, m_1 to m_{15}, which are denoted by solid circles. And m_8 is the query generator who send the request to form a cloaked region. Other solid users are receiving the request from m_8. The dotted circles represent users' wireless communication range., and the arrow represents its maximal moving speed. Generally, the cloaking region construction will consist with the following three phrases:

Phase 1: Peer searching phase The query issuer m_8 wants to form the cloaked region to blur the location information of its query. m broadcasts a **FROM_GROUP** request message with a message sequence ID and the hop distance h to its neighbors. Then, m_8 will listen to the network and wait for the responses.

When a peer gets a **FROM_GROUP** request message, it will first check the message sequence ID to see if it is a duplicated message. If it is a duplicated message, the peer will simply reply an **ACK** message without further processing. Otherwise, the peer will process the message based on the h:

Case 1: $h = 1$. It means that the peer will be the last hop for this query request. The peer who receives the **FROM_GROUP** request message will not broadcast it further. The peer will return its ID, current location and the maximal moving speed to the query requester.

Case 2: $h > 1$ The peer will decrease the h value and further broadcast the message to its neighboring peers. After it collects all the response from the neighboring peers, it will return the set of the peers' information(ID, current location and the maximal moving speed) to its parent peer. The query issuer will collect all the response from the neighboring peers, and count them to see if there are at least $k - 1$ peers in the

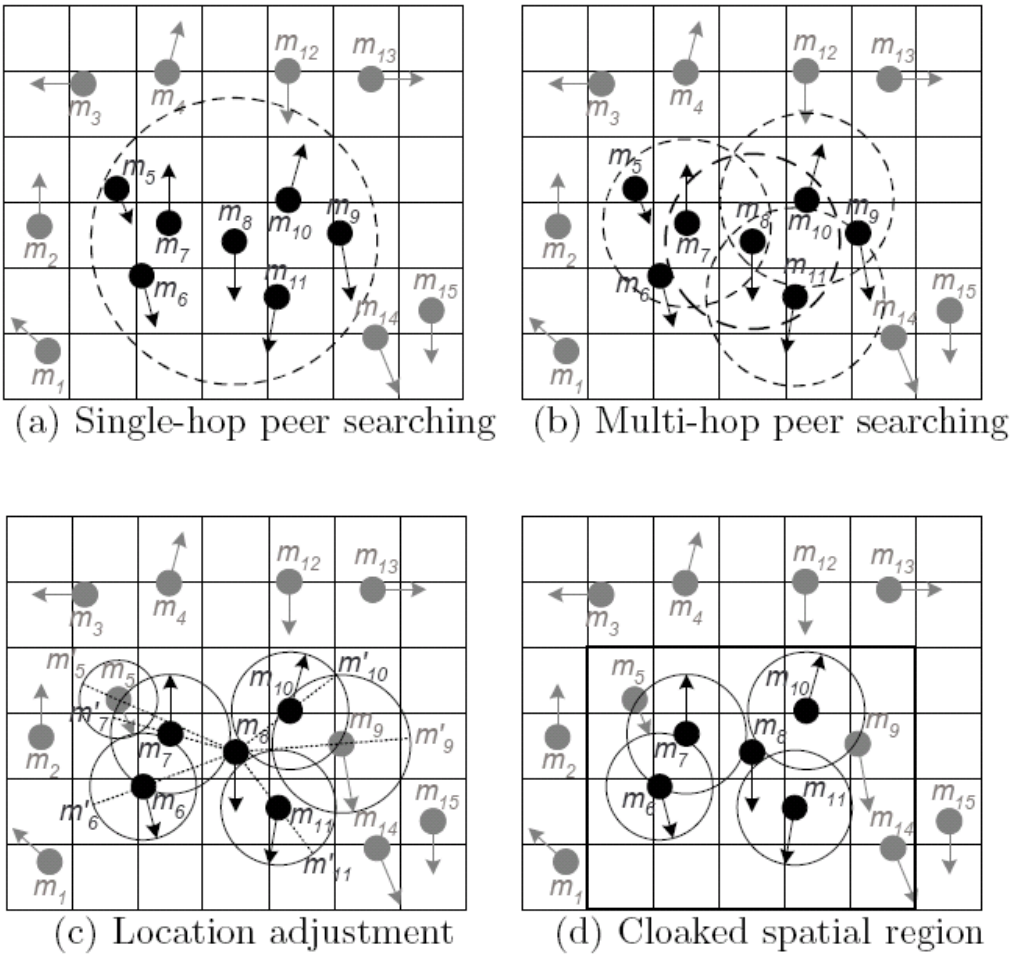


Figure 3.7: P2P spatial cloaking algorithm. [4]

result set. If there is not enough peers in the set, the query issuer will try to re-broadcast the **FROM_GROUP** request message with a larger h value. The query issuer will continuously increase the h value until it gets the responses from at least $k-1$ other peers.

Figure 3.7a and 3.7b depict single hop-and multi-hop peer searching example, respectively. In Figure 3.7a, the query issuer,m_8, ($k=5$), finds enough peers in one single wireless communication hop. On the hand, as shown in Figure 3.7b, the query issuer m_8 can only find 3 other peers directly. So m_8 have to increase h value. And m_7, m_{10} and m_{11} will re-broadcast the **FROM_GROUP** request message with a decreased h value.

Phase 2: Location adjustment phase Because in the authors' assumptions that the peers in the system are moving, we should take the possible movement of the peer in consideration. Otherwise, at the time that the cloaked region some of the peers may not in the region any more, which is against the K-anonymity principle and will lead to a privacy leakage issue. As the consequences, we will calculate the greatest possible distance by an equation, $|mp'| = |mp| + (t_c - t_p) \times v_{max_p}$, where $|mp|$ is the Euclidean distance between m and p at time t_p, i.e., $|mp| = \sqrt{(x_m \times x_p)^2 + (y_m \times y_p)^2}$, t_c is the current time,t_p is the timestamp while v_{max_p} is the maximum speed of p. The cloaking algorithm will use v_{max_p} to decide the possible location for p, in order to decide the final cloaked region.

Figure 3.7c illustrates the situation, for each black dotted points, the circle represents the possible location the peer can be at time t_c which is calculated by their maximal travel speed. The greatest possible distance between the request originator m_8 and the neighboring peers, m_5, m_6, m_7, m_9, m_{10}, or m_{11} is illustrated by the dotted lines. For example, the distance of the line $m_8 m_{11}^{'}$ is the greatest possible distance between peer m_8 and m_{11} at time t_c, i.e., $|m_8 m_{11}^{'}|$.

Phase 3: Spatial cloaking phase In the cloaking phase, the query issuer, m, will firstly form a group with at least *k-1* neighboring peers, based on the greatest possible distance which is decided by the location and the maximal speed of the peer. The preliminary cloaking region will be constructed to cover the minimal possible area for the *k-1* other peers. If the area of A is less than A_{min} which is described in user's privacy profile. The algorithm will extend the preliminary cloaking region, until it satisfies A_{min} requirement. Figure 3.7c gives a clear example, the $k-1$ nearest peers include m_6, m_7, m_{10}, and m_{11}, while the query issuer is m_8. For instance, the privacy profile of the query issuer, m_8, is ($k = 5$, $A_{min} = 20$ cells), and the required cloaked spatial region of m_8 is demonstrated by a bold rectangle, shown in Figure 3.7d.

After the cloaked region is constructed, the original query issuer will randomly pick a peer in the group to send the query to the service provider as its agent, in order to hide the its identity.

3.3 Summary

Both the centralized solution and the decentralized solution have their advantages and disadvantages.

The centralized spatial cloaking

Advantages: The most significant advantage is that the location anonymizer have the users' location information in the entire area, it is able to form the K-anonymity group and construct the cloaked region effectively and efficiently. And the centralized solution is very adaptive that it is very easy for it to support range query, nearest neighbor quer and other kinds of spatial query type, as long as the service provider have the privacy-aware processor embedded. Moreover, the users don't need to worry about the information leaking at the service provider side.

Disadvantages One potential problem for these kind of solutions is the liability of the location anonymizer. Because every query requests will go though it, it will not only deal with the spatial cloaking for each query request, but also it will keep updates for the location information of the users. It will easily become a performance bottleneck. And it is a single trusted party in the system which will make it a obvious attacking target and vulnerability for the system. And also some other studies and researches done by Chow and Mokbel [3] point out that the centralized solution may not very strong to defend the continuous query attack.

The decentralized spatial cloaking

Advantages The major advantage for the decentralized spatial cloaking is that they don't need the trusted third party, the location anonymizer any more. It can prevent the performance bottleneck as well as the single point of failure.

Disadvantages One of the obvious problem is that there are much more communication overhead in the process of cloaked region construction. As the result, it may cost more time to construction the cloaking region comparing with the centralized solution. And the assumption for these kind of solutions assume that the users in the system are trustable and willing to cooperated with each other to constructed the cloaking region. If there are some malicious users in the group, the privacy leakage possibility for the users will be quite significant. The last but not the least, the current cloaking algorithm is also suffers from problem that the actual query issuer will always be the center portion of the cloaked region which is due to the broadcasting communication nature.

Chapter 4

Cache Management Techniques for Privacy Preserving Location-based Services

4.1 System Overview

From the perspective of queries, privacy is protected only for a single snapshot location-based query in these aforementioned systems. Users are not protected from query tracking attacks and correlation attacks [1], [3]. For example, if a mobile user launches the same query from different locations (i.e., continuous query), the mobile user's location can be identified by comparing the users in all the related cloaked regions.

Our cache management techniques can effectively decrease the number of queries which has to be forwarded to service providers and successfully alleviate correlation attacks. Location-Based Services can be generally defined as services that integrate the location of mobile devices with other information so as to provide added value to mobile device users [14]. For example, a motorist can find his/her closest gas station through LBS when he/she drives on a highway. LBS have been well developed during the past decade and many of them are very popular in our daily lives (e.g., navigation services, friend finding services [5], etc.). However these prevalent location-based services could be a potential threat to user privacy. Consequently both location-based service providers and mobile users should be careful and sensitive regarding the way location related information is handled. In addition, governments (both the U.S. and EU) have also legislated regulations on the usage of personal location information [14].

In this chapter, we describe our cache management techniques for supporting privacy preserving spatial queries in mobile environments. The fundamental idea behind our

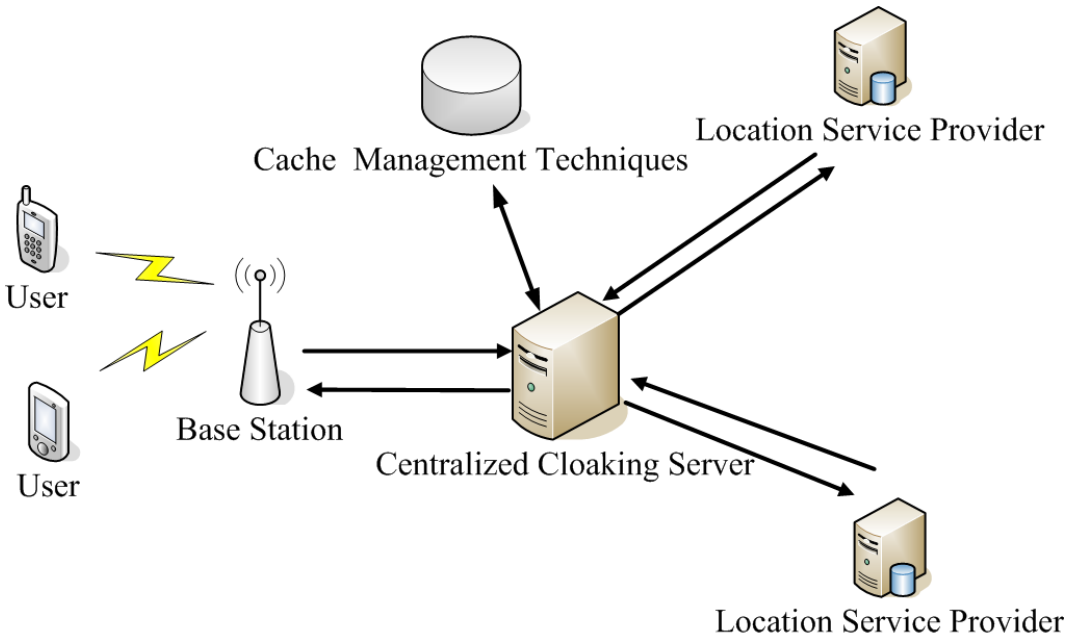


Figure 4.1: System Architecture

methodology is to leverage the cached results from prior spatial queries for answering future queries at the location anonymizer.

4.2 System Architecture

Our system architecture is consisted of four main entities: mobile users, the location anonymizer, cache management techniques, and location-based service providers as illustrated in Figure 4.1. We consider mobile clients such as cell phones, personal digital assistants (PDA), and laptops, that are equipped with global positioning systems for continuous position information. In addition, we assume that there are access points/base stations around the system environment for mobile devices to communicate with the location anonymizer. All users are mobile and travel on underlying road networks.

The location anonymizer is an intermediate agent which can be trusted by mobile users. The location anonymizer receives continuous location updates from mobile users and stores their locations with an index structure. In addition, the location anonymizer also anonymizes the location of any query requesting mobile user to a cloaked region before forwarding the query to related location-based service providers. Any user identity related information in the query is also removed by the location anonymizer during the cloaking process.

Location-based service providers play the role of spatial data maintainers and spatial query processors in our system. In order to handle privacy protected spatial queries, location-based service providers implement privacy protected query processors in their databases. The privacy protected query processor has the ability to process cloaked spatial queries efficiently and retrieves the inclusive result set (i.e., the minimal set which covers all the possible answers) for query requesters. After receiving the result set, mobile users can distill the exact answers from their locations in linear time [10]. Basically, strict privacy requirements increase the complexity of processing a location-based query.

4.3 Cache Based Spatial Query Processing

Caching is a key technique to improve data retrieval performance in mobile environments [9]. As we can see in Figure 4.1, all the spatial queries and returned query results have to pass through the location cloaker. Consequently, if the location anonymizer can cache the received query results from service providers, the cached results can be utilized to fulfill new spatial queries from mobile users. By applying this cache based solution, mobile users privacy protection can be further improved. Since the location anonymizer can solve a certain number of queries without forwarding them to service providers, it would be much more difficult for adversaries to launch correlation attacks.

For each received spatial query result, the location anonymizer calculates the minimum bounding rectangle (MBR) of all the returned spatial data objects. Then, these retrieved data objects will be inserted into the cache and the boundary of the cached region with be

adjusted based on the MBR. Figure 4.1 demonstrates the relationship between the cached region (the shaded area) and the whole search space. Since k nearest neighbor (kNN) query and window query are two common types of spatial queries, we focus on the two spatial query types in this paper. We introduce our cache based spatial query processing techniques as follows.

4.3.1 *k* Nearest Neighbor Queries

For kNN queries launched by mobile users, the location anonymizer first checks if the query point is covered by the cached area. If the query point is covered by the cached region (e.g., points A and B in Figure 4.2), the location anonymizer will try to retrieve k cached objects to answer the kNN query. Basically, there are two possible conditions - the query can be totally fulfilled or the query can only be partially fulfilled. The mobile user at point A requests for three nearest points of interest (POI) and we can retrieve three nearest objects from the cache based on their spatial relationships. Therefore, the query of mobile user A can be solved without forwarding the query to any service providers. Similarly, the location of mobile user B is covered by the cached region and its k value is equal to two. For this query, we can only retrieve one POI P1 whose verification circle (with the distance between P1 and B as the radius and B as the center point) is totally covered by the cached region. POI P2 demonstrates a counter example. Since we are not sure if there is any POI within the non-cached region outside the cached area, P2 cannot be count as a nearest neighbor of B [11]. Consequently, the kNN query of B still needs to be forwarded to service providers. However, the partial result (i.e., POI P1) can be returned to mobile user B for decreasing the response time. If approximate results are acceptable, POI P2 will also be returned. For mobile user C, since its location is out of the boundary of the cached region, the location anonymizer will forward its kNN query to service providers. A B C C Figure 4.2.

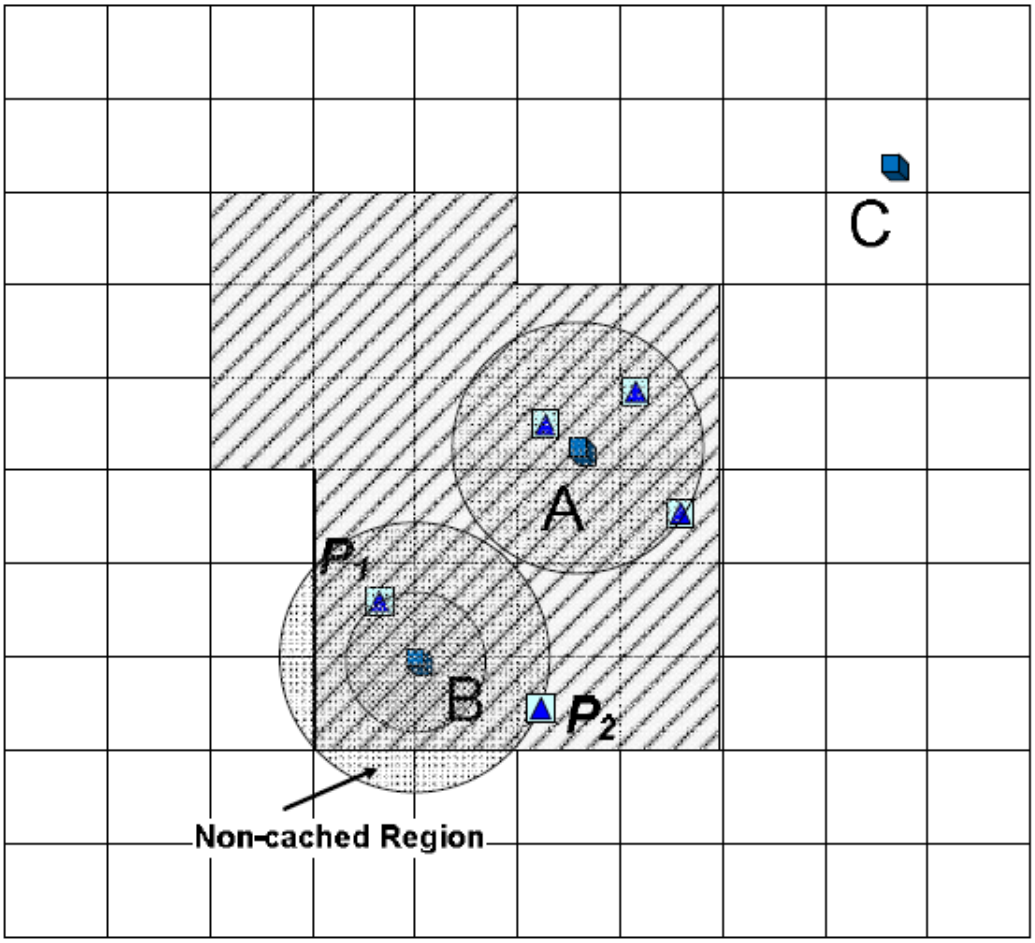


Figure 4.2: kNN query examples.

4.3.2 Window Queries

Window queries find data objects within a specified area - the query window. Generally, there are three possible spatial relationships between the query window and the cached region. First, the query window is totally cover by the cached area (e.g., query window B in Figure 4.3). The query can be directly answered by the location anonymizer without forwarding it to service providers. Second, the query window is partially covered by the cached area (e.g., query windows C and D in Figure 4.3). For this condition, the location anonymizer still needs to forward the reduced query window (the portion not covered by the cached area) to service providers. However, both the query processing time and communication costs can be effectively decreased. Third, there is no overlapping between the query window and the cached area. Consequently, the cached data set cannot be applied to improve query evaluation performance and privacy protection.

4.4 Cache Space Management and Replacement Policies

In order to improve the performance of our cache based solution [17] (i.e., cache hit rate), we have to develop efficient cache space management mechanisms. As illustrated in Figure 4.2, the cached data in the location anonymizer is indexed by a grid structure. Based on statistics, the query frequency of each POI type is not uniformly distributed during a day. For example, there are significantly more queries for restaurants during dinning time (i.e., noon and evening). The frequency of queries for gas stations increases during rush hours and there are more queries for hotels in the evening. According to the variation of query frequency for different POI types, we design a temporal dynamic cache space allocation mechanism as illustrated in Figure 4.4. We verified the feasibility of our design with extensive simulations and the results are presented in Section 4.

For cache replacement policies, we apply three methods to decide which grid cell should be replaced based on time, retrieval frequency, and mobile user density.

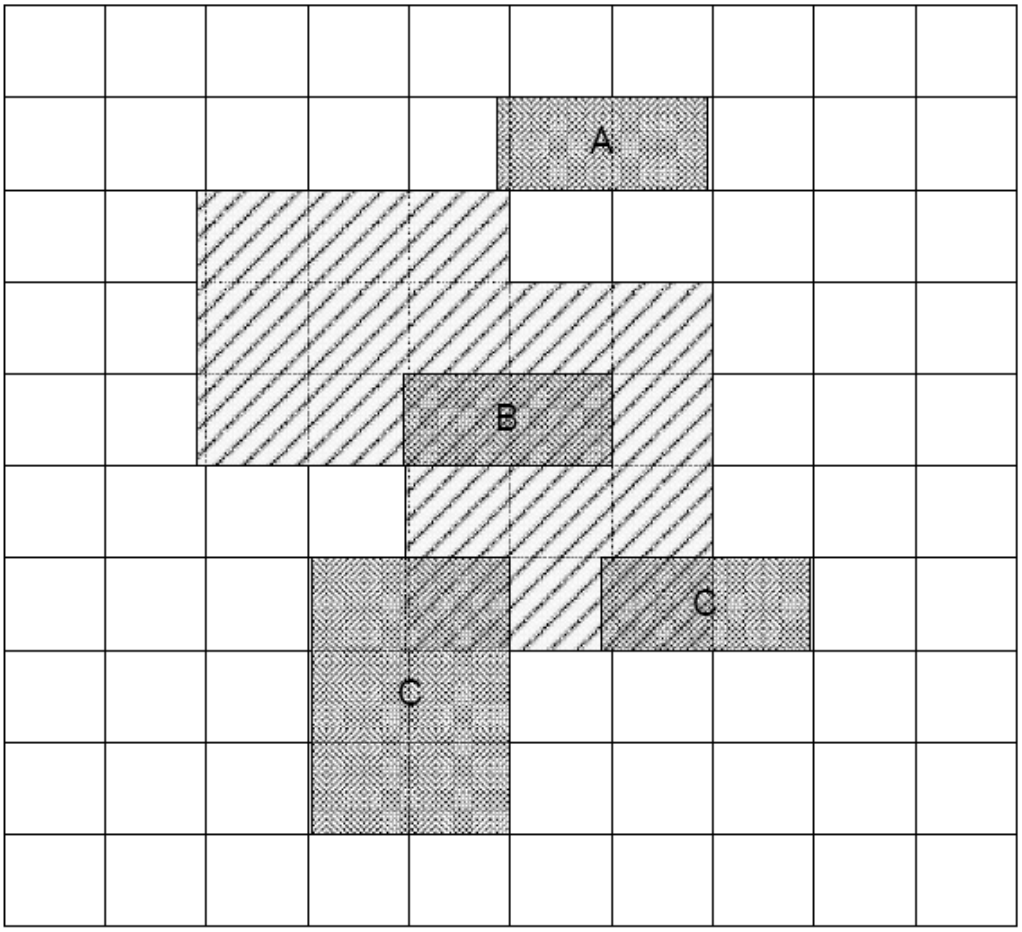


Figure 4.3: Window query examples.

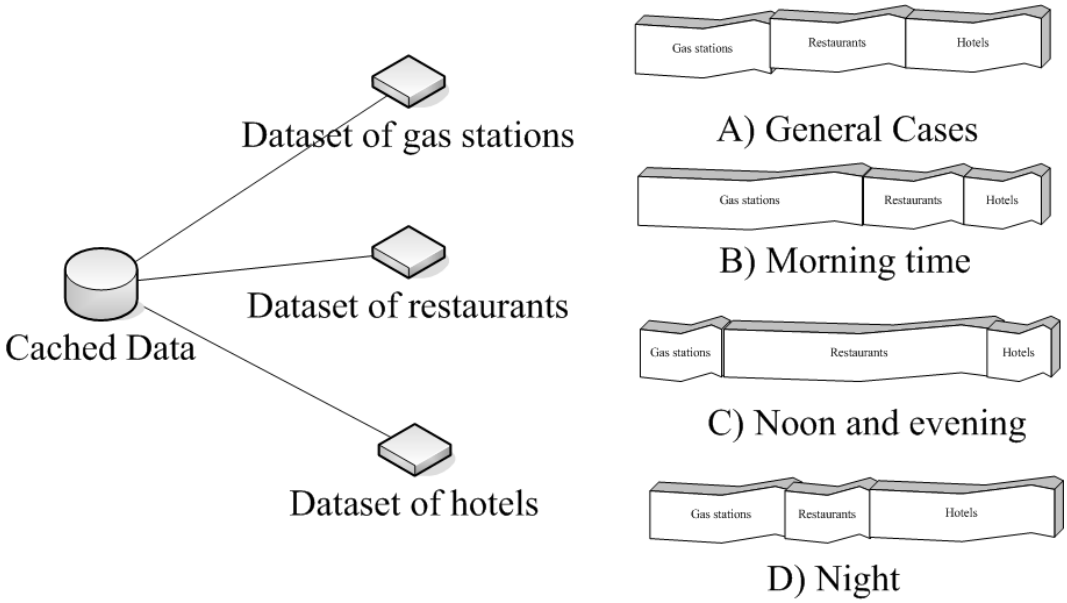


Figure 4.4: Dynamic allocation of cache space based on spatial query frequency.

Time Based Policy The weight of each grid cell is according to a timer, which records the time interval from the last visit to present. Similar to the Least Recently Used (LRU) algorithm, the cell, which has the largest time interval, will be discarded first.

Retrieval Frequency Based Policy Since retrieval frequency reflects the popularity of a certain data object/spatial region, this method decides the weight of each grid cell based on the number of times which it has been searched. The cell with the lowest visit frequency will be replaced first.

Mobile User Density Based Policy Mobile users usually interest in POIs close to their current locations. Accordingly, it is an ideal strategy to keep grid cells which have high mobile user density and discard low user density cells.

4.5 Experimental Results

To evaluate the performance of our approach, we have implemented our cached based query solutions and cache management mechanisms within a simulator. The objective of our design is to decrease the number of queries which have to be forwarded to service providers to preserve mobile users privacy, save computational power, and decrease communication costs. Based on our novel cache replacement policies, the cache hit rate can be effectively increased.

4.5.1 Simulator Implementation

Our simulator consists of four main components, the mobile environment, the location anonymizer, the cache management module, and the location-based service provider. For the mobile environment, we applied the network-based moving objects generation framework [1] to generate a set of mobile users and the underlying road network inside the city boundary of Oldenburg in Germany. Each mobile user is an independent object which encapsulates all its related parameters (e.g., its current speed and destination). We implemented our

query processing and cache management techniques as new modules for interacting with mobile users to improve query performance and privacy protection.

Every simulation has numerous intervals (whose lengths are Poisson distributed), and during each interval, the simulator selects a random subset of mobile users to launch spatial queries (the query intervals are also based on the Poisson distribution). The subset size is controlled by the user defined mean number of queries per minute (e.g., 1000 queries per minute).

To obtain results that closely correspond to real world conditions, we obtained our simulation parameters from public data sets, for example, mobile user and gas station densities in Oldenburg.

Mobile Users: The population in Oldenburg is 159,282 based on Wikipedia. According to the mobile device penetration rate in Germany, we estimate that there are around 5,000 mobile users served by one location anonymizer.

Points of Interest: We obtained the information concerning the density of the of interest objects (e.g., gas stations, restaurants, etc.) in Oldenburg from Google Maps. Because gas stations are commonly the target of spatial queries, we use them as the sample POI types for our simulations. According to Google Maps, there are 1,399 gas stations inside the city boundary of Oldenburg.

4.5.2 Performance of the *k*NN Query

We first tested the performance of our three cache replacement policies with k nearest neighbor query. We increased the number of queries per time interval from 1 to 5000. As we can see in Figure 4.5, the cache replacement policy based on mobile user density prevails over two other strategies. The cache hit ratios of our two novel replacement policies are remarkably higher than the traditional LRU solution.

Figure 4.6 shows the effect of our temporal dynamic cache space allocation mechanism. Our technique improved the cache hit rate for one time interval, Noon & Evening. However,

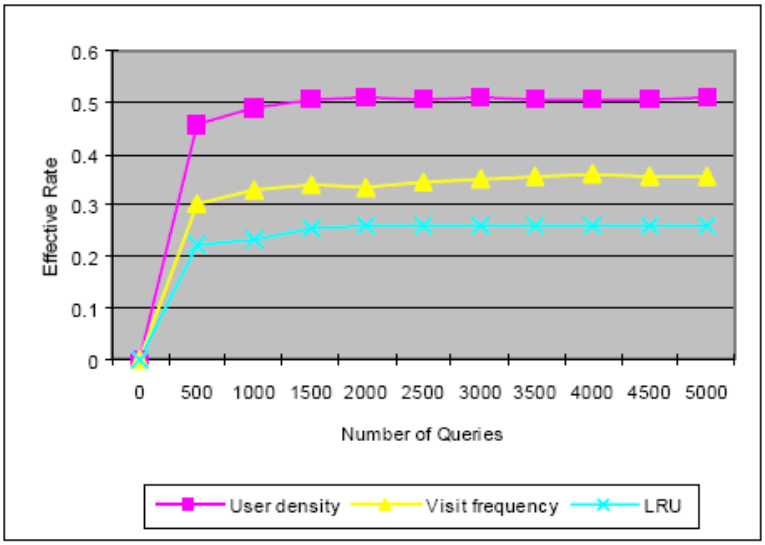


Figure 4.5: The cache hit ratio of the three cache replacement policies with increasing kNN query number.

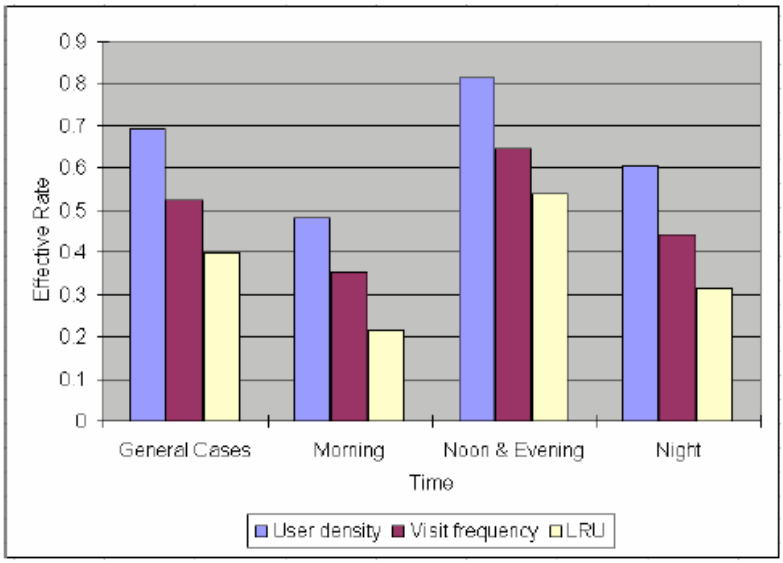


Figure 4.6: The cache hit ratio of different time intervals during a day with our dynamic cache space allocation mechanism.

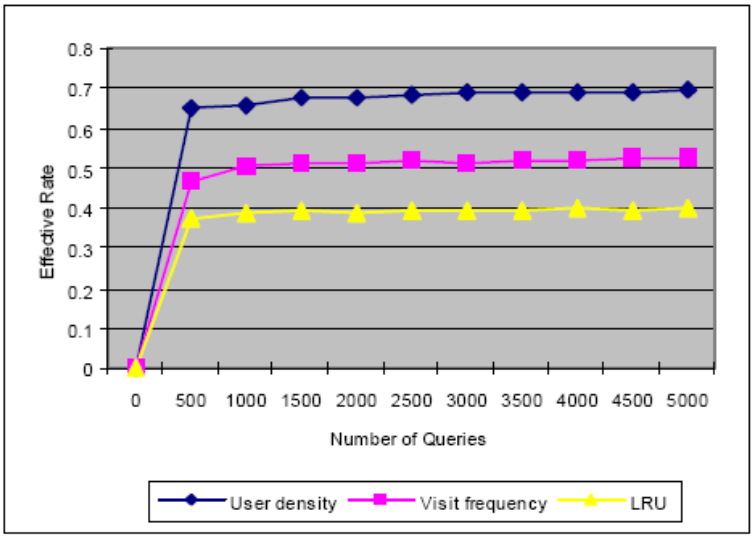


Figure 4.7: The cache hit ratio of the three cache replacement policies with increasing window query number.

there was no improvement in other two time intervals. Since mobile users behavior varies at different locations, users may decide when to apply our mechanism based on statistics and experimental results.

4.5.3 Performance of Window Query

To see the effect of our cache replacement policies on window queries, we increased the query number from 1 to 5000 and the result is demonstrated in Figure 4.7. Similar to kNN query, the cache replacement policy based on mobile user density outperforms two other strategies and the performance of our two solutions are better than LRU.

We also studied the effect of various query window sizes by enlarging the query window size from 0 to 1/100 of the whole search space and the results are shown in Figure 4.8. Basically, the result trend is very similar to the previous experiment.

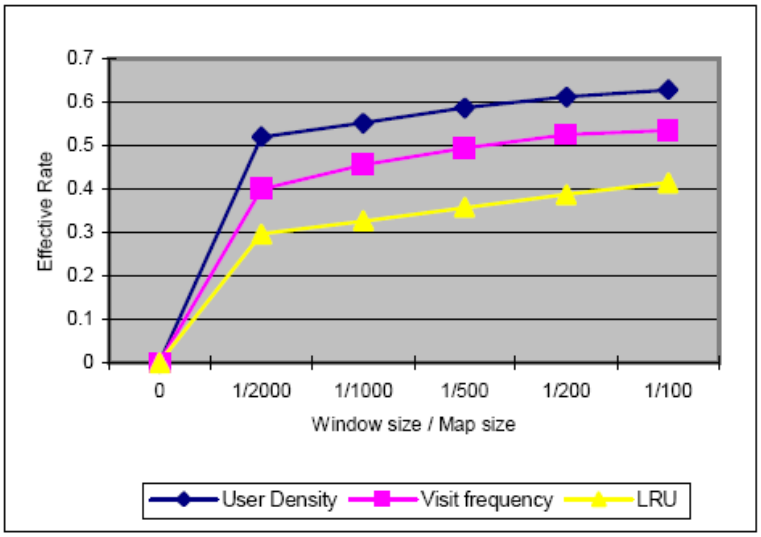


Figure 4.8: The cache hit ratio of the three cache replacement policies with increasing query window size.

Chapter 5

Road Networks based Spatial Cloaking

5.1 System Overview

The conventional frameworks and solutions only focus themselves on how to find enough peers in the neighborhood to construct the cloaking region. The importance of the indistinguishability of these users is ignored. In the existing solutions, the spatial cloaking algorithm treats all the users in the same way. It is true if the attackers only have the information about the user's location. However, it is the case in the reality. In the most attacking scenarios, the attackers are always able to obtain additional information about the targeted area. Combining this area information, the attacker can easily exclude the users who are very unlikely to issue the query(which indicates the query issuer) in the cloaked region. And in that way,K-anonymity principal will be compromised, and the user's location privacy will be suffered.

The formal definition of the K-anonymity, the idea is firstly proposed in the data privacy techniques. A release provides K-anonymity protection if the information for each entry in the release set cannot be distinguished from at least *k-1* individuals whose information also appears in the release [15].

K-anonymity in the location privacy is quite similar with it is the data privacy. We try to hide the actual query requester within *k-1* other users. In order to achieve the K-anonymity there are two criteria: first, there must be *k-1* other users to satisfy the privacy in terms of anonymity quantity. Secondly, it is the quality of the anonymity, which is the indistinguishability of these *k-1* users. In the original formal definition, these k entries (including the one we want to protect) should not be able to tell the difference and identify.

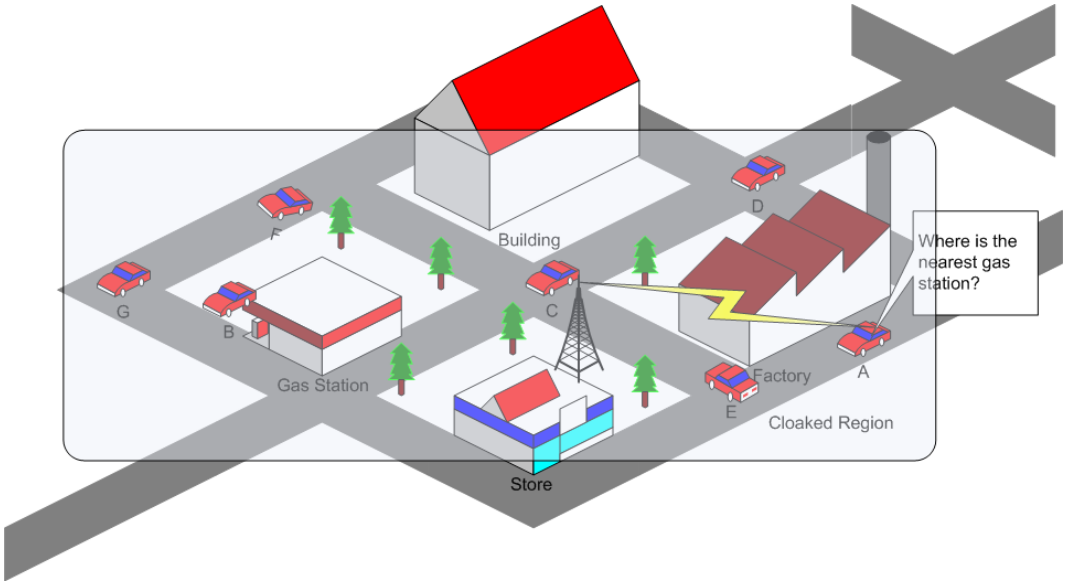


Figure 5.1: Example of extra information attack.

A common example is that the attacker may know all the local POI information. And a more extreme example is like this: the attacker not only can obtain local POI information but also can record the trace of individual user (the attacker can only record the route the user travels but cannot identify the identity for each user). For example, the one user issues a query about "*where is the nearest gas station?*".

As shown in Figure 5.1, assuming that K value for this query is 7, the trusted server will search for 6 other peers to construct the cloaked region represented by the light gray box. But if the attacker has the ability to obtain the local POI information and/or have the ability to monitor the query area as described above. After mapping the cloaked region with the actual map, the attacker may easily notice that 4 of these 6 users (user B, C, G and F) are very close to a gas station that very possibly they are not going to issue this

kind of query right now. The actual K value in this scenario will be degraded to 3, which do not satisfy with user's privacy requirement. Moreover, if the attacker has the ability to track the historical information, he might further discover the other 2 of them (user D and E) have been the gas station not long ago. In that case, the attacker will easily figure out that user C is the real requester of this query.

In this chapter, we present a framework which can easily solve this kind of vulnerabilities in location privacy protection by using an incremental based cloaking region construction method. Also, we will apply the idea in a peer to peer model to avoid the single failure/attack point and performance bottleneck problems. Moreover, the research work will extend the framework to the road networks which will give a more precise spatial query result to the user. And peer management and the cloaking region construction are also integrated in this framework.

5.2 System Architecture

Assumptions:

- Clients have the capability to communicate with the service provider.
- Clents also can talk with each other via the wireless connections.
- Clients a GPS module to get its location.
- Clients follow the protocol and are willing to work corporately.
- Clients have a limited storage on board that it cannot hold all the map information. But it has the ability to store the local map in their devices. So every client will have the map with the same pattern of map partition. And based on the partition, we will have a tree constructed to organize these small blocks in the map.
- Entries in the tree at any level represent a small communication cluster, and a client will be selected as the cluster head to manage the cluster. The cluster head will maintain the information about the total number of the client nodes it manage and

their exact locations which will be used in the refine phrase of the cloaking region construction algorithm.

- The internal passing messages will be routed based on client's ID, using GPSR liked geographic based routing protocol.
- The client do not have identities in the network only client's ID information which they calculated them based on the CAN partition algorithm and its location information.

5.2.1 Road Network Partition

The partition algorithm is trying to organize and index the road map using a CAN based method. First of all, the map will be abstracted into intersection points with a weight. The weight value will be decided by both the degree of the intersection which is number of road segments crossed and the total length of these road segments.

After that, a threshold will be decided to divide the space which is used to constrain the maximal weight can have in the basic unit. The partition algorithm will follow the CAN pattern, and we divide the space by half every time until the weight in the cell is below the threshold and cannot be divided further. In that way, the whole area map will be divided into small blocks which are equivalent in weight. And meanwhile, a binary tree will be constructed accordingly, as illustrated in Figure 5.2.

5.2.2 Peer Management Operations

Each entry in the tree structure represents a communication cluster. And the one of client inside the cluster will be selected as the cluster head. And the peers in the whole space will be organized into a tree as shown in Figure 5.3. The cluster head will be responsible to store and maintain the information about total number and the location of the clients within its cluster.

And the basic operations for a client in the scheme are listed as follows:

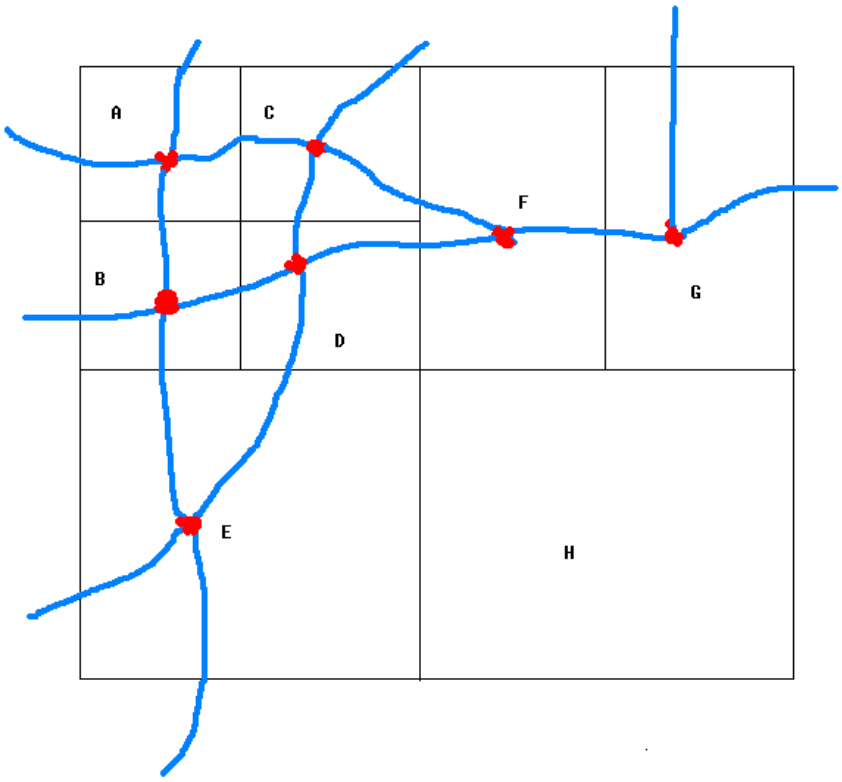


Figure 5.2: Example of CAN based Road Partition.

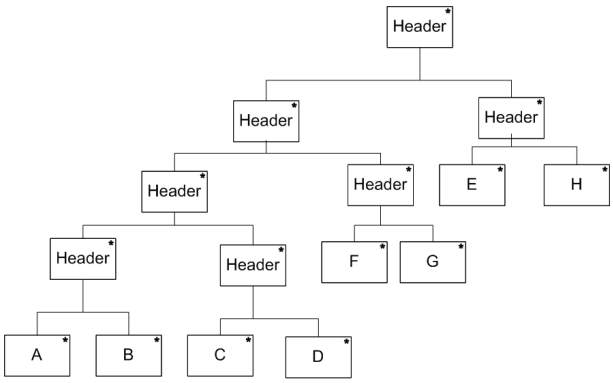


Figure 5.3: Example of CAN based Peer Management.

New Client Join The new client will broadcast to find the clients that already in the block. If there is already have the clients in the current cluster, which means there is already a cluster head for the block. It will pass the join information to the cluster head, and the head will update the information about the number of the clients it managed and forward it to its parent client in the tree. If it didn't find the cluster head there which can be confirmed by the fact that no other clients in that block response, it will automatically claim it is the cluster head for that. And it will recursively send that information to the upper level region until get the response from some cluster head. It is possible that the available cluster head is out of the communication region. So if any of the clients will help the new client to find its cluster head based on its known information.

Client Leave If the client is not the cluster head it will simply inform the cluster head, and the head will update the corresponding group information. If it is the cluster head and have more than itself in the cluster, it will choose a new one pass all the information to it and make it as the new cluster head. And it will notify its cluster

head as well. If it is the only client in that block, it will notify its parent client which will take the cluster head for the leaving part.

Client Move Because the client should moving between some quite close regions. So it is possible for us to make two cluster head talk with each other to make the update for the client movement. When a node will move out of its current block in a short time, it will notify the current cluster head and the future cluster head. In that way, the update information and its communicational overhead will be reduced significantly.

5.2.3 Cloaking Region Construction

When a client wants to issue a cloaking construction, it will issue the parameter K which indicates the number of the peers the client want to have in the cloaking region and the parameter L which indicates the minimal road length requirement in the cloaking region. After that, the client will first communicate with its cluster head to check if the current cluster has the enough clients in it. If it has, the cloaking region will be constructed on that block. And then it will check if the minimal road length requirement is meet. If not, it will recursively ask the upper level cluster head following the tree structure until the head gets the enough clients and meet the length requirement.

After the initial cloaking process completed, the client will send the region using the very basic level blocks in a random order to the service provider querying for the POI, one block a time. Every time the client sends one block to the service provider and gets the corresponding result about the POI information. And the actual cloaking region is constructed in the incremental way, if the unqualified clients present. It will compare the POI location with the location of the users, if the user is too close to the POI the user will be excluded because it is not a qualified peer in the K anonymity definition. For example, the clients who are in the gas station are not qualified in the nearest gas station queries. The process will continue until the client gets enough number of qualified candidates.

And if it happens, the actual cloaking region will move up one more level in the tree. And it continues to do this, until it finds the enough qualified users satisfied the

K anonymity. In that way, the attacker cannot distinguish real block which contains the query issuer and the redundant blocks.

5.3 Incremental Spatial Query Process

Lemma: If the result set includes all the edge points' K nearest neighbor and the POIs inside the cloaking region, every inside points' K nearest neighbor will be included in the result set.

Proof: Assume we have point P inside the cloaking region and its Kth nearest neighbor K is not included in the result set. And K is outside of the cloaking region. So the shortest path between P and K, D_{PK}, must come through an edge point E of the cloaking region. However, K is not included in E's K nearest neighbor set. So the distance between K and E must be greater than the upper bound of E's K nearest neighbors. So there must be a better candidate inside E's K nearest neighbor set or there is another shortcut between P and K which is not via E. Both of them are both against the assumptions.

5.3.1 Algorithm Description

Search Bound It is the search upper bound for each queue. It is decided by the distance of its Kth nearest neighbor in the POI candidate list. If the Search Distance meets the Search bound, the search process for this queue or node is end and the queue will be destroyed.

Search Distance It is the distance of the first item in the priority queue. It reflects the searched area for a particular node. It is the minimal distance from the segment to the search start point.

Search Threshold is the distance decided by the user to determine when to stop the current search and move to the next queue. The Search Threshold will be updated every round.

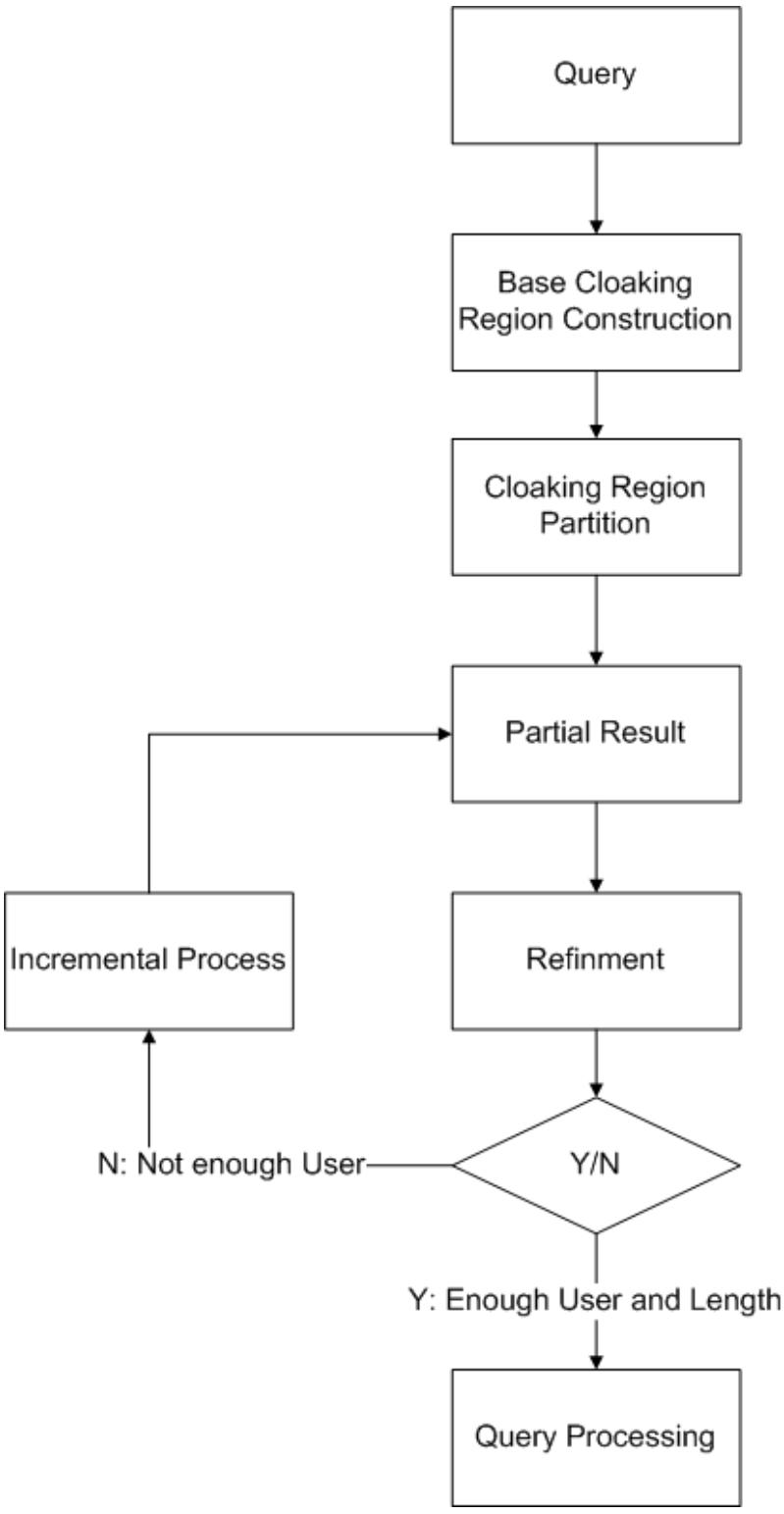


Figure 5.4: Incremental cloaking and query process.

The basic idea for this search mechanism is that for each edge points in the cloaking region we will do a K Nearest Neighbor search to guarantee that the result set will include all points' K Nearest Neighbor.

A priority queue will be constructed for each edge points based on the distance between it to the searching point. And the search will be executed like the round robin scheduling mechanism in Operating Systems. Every edge point will execute a K nearest neighbor query partially and turn by turn until the Search Threshold for each round is met. When the current searching distance is equal or more than the search distance for the current round which is the distance of the first item in the priority queue, the search in this priority queue will be stopped and the process will be continued to the next priority queue.

There will not be an upper bound if the number of the objects in the POI list is smaller than k, until it has at least k POIs in the candidate list. If the number of the objects in the POI list is equal or greater than K the upper search bound will be the distance of Kth nearest neighbor to the searching edge point. And every time a POI is discovered and added in the list the bound will be updated and the search stops when the search distance is equal or greater than the bound value.

If the searching distance is greater than the current upper bound, the searching process for this edge point will terminated. And the queue will be destroyed. A search is completed if all of the searching priority queues are destroyed.

And all these searches at the edge points will share a POI list and a discovered node list.

For the POI list, every time POI was found it will be inserted into the list and calculate the shortest distance to each edge points to determine the searching bound for each searching priority queue. If the searching distance on that side can be stopped. For example, initially, *POI1* is in this list and the distance from *POI1* to *E, I* and *K* will also be calculated and stored in the list.

And the discovered node list is used to guarantee that each search of the edge points do not have any overlaps which will be an overhead or waste for the search resources. Each

Algorithm 2 Incremental Road Network based K Nearest Neighbor Searching

Input: Map M, Discovered Node List DL, POI set PS, Edge Points set EP, K value K, Search Threshold ST

Output: set of POI candidates

```
1: Construct a Search Queue List QL
2: for each Edge Points in EP do
3:    Add a new priority queue into QL
4:    Update the POI distance to each edge points
5:    if the number of the POIs in PS ≥ K then
6:       Update search bound to the Kth nearest POI
7:    else
8:       set the search bound to ∞
9:    end if
10: end for
11: Sort the queue list by their search bound
12: while QL is not empty do
13:    for each priority queue do
14:       if Is the last queue in the QL then
15:          Update ST
16:       else
17:          while True do
18:             if Search Distance ≥ Search bound then
19:                End search in this queue and destroy the queue
20:             else
21:                if Search Distance ≥ Search Threshold then
22:                   Jump out while and move to the next queue
23:                else
24:                   Search the first road segment in the list {Algorithm 3}
25:                   Insert the adjunct segments {Exclude the discovered nodes}
26:                end if
27:             end if
28:          end while
29:       end if
30:    end for
31: end while
32: Return POI candidate set
```

Algorithm 3 Inserted Road Segment Check

```
1: Check the inserted segment
2: if one of the node in the discovered node list then
3:    Return {The segment is searched previously}
4: else
5:    if POI found in the segment then
6:       Insert POI into POI set
7:       Update searching bounds
8:    end if
9: end if
10: Check the searched point
11: if all of its neighbor is searched then
12:    Insert into Discovered Node List
13: end if
```

time when the searching process is executed, it will first take a look at if current segment contains the node inside the discovered node list. If so, this segment will be popped out of the queue directly, not have it read into the memory. For example, at the beginning of the search point J is in the discovered node list. And if IJ is in one queue, the check segment process will not take it but pop it directly out of the queue.

5.3.2 Example of Incremental Spatial Query

Scenario: The cloaked region is showed in the figure bounded by the yellow nodes. And the K value for this query is 2.

Assumption: The sorted order for the edge points is E, K and I. And the search threshold is 10 and increased by 10 each time.

As shown in Figure 5.5, when the search begins for each point at the edge of the cloaking region, we will construct a queue for them. As shown in figure. There will be queue E, queue K and queue I. The discovered node list will include J and the POI list will have POI 1 in it.

The search starts from queue E. it will first insert ED, EJ and EF into the queue. And we perform the search process in ED and EF. There is no POI there, and we meet the

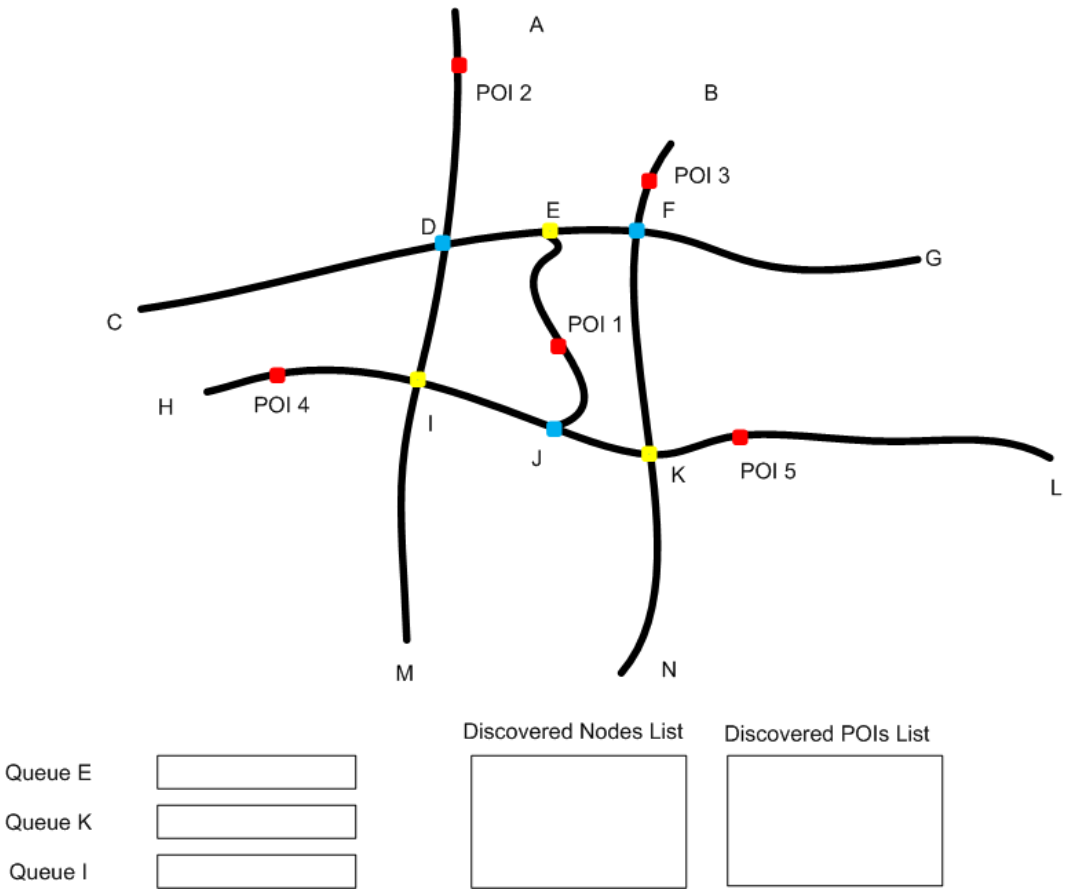


Figure 5.5: A running example for the incremental spatial query.

search threshold 10. And we will move to the next queue, K. And queue E will have BF, FG, AD and CD in it. And E will be inserted into the discovered node list.

For K, FK, LK, and KN will be inserted. The search order is FK, KN and KL. We don't find any POI in FK and KN. But we find POI5 in KL and insert it into the POI list. Meanwhile, we will have the search bound for K now, which is 15. And we find that the current search distance is greater to the search bound. The search process for K will be ended and the queue K will be destroyed. And K will be inserted into the discovered node list. And the search will continue to queue I.

For queue I, we have the search bound 35 which is the distance between POI5 and I. and DI, HI and IM will be inserted and checked. We find POI4 in IH, and inserted it into POI list. The search bound for I is updated to be 20. But the D may have better candidate, and we meet the search threshold 10. So we insert I into the discovered node list, stopped the search here now and move back to queue E.

And the search threshold is updated to be 20.

In queue E, the search bound is 35 which is the shortest distance between POI5 and E. And BF is checked and we find POI3 and updated the search bound for E to be 15. But the search distance now is 10 which is smaller than search bound and search threshold. We check FG AD and CD sequentially. We find POI2 and insert it into the POI list, but the search bound is not changed. And search distance now is greater than the search bound, so the search for E will be ended.

And we check the only remained queue I. We find D is in the discovered node list and pop its segment and pop it out. The search distance now is also greater than the search bound. K will be destroyed.

And for all the queue in the queue list is destroyed, our search is terminated and POI list will be returned.

5.4 Simulator Implementation

We implemented the technologies and algorithms mentioned above which protect location privacy by peer-to-peer based cloaking on road networks. We name the prototype system as PROS. With PROS, a mobile user forms a cloaked road segment set by collaborating with her peers when she needs to retrieve information from location-based service providers. Afterward, the cloaked road segment set is sent to the service provider for query processing and an inclusive query result set is returned to the requestor after the query evaluation.

5.4.1 Cloaking Phrase

Cloaked Road Segment Set

PROS employs road segment sets to represent the cloaked region to achieve advanced location privacy protection effect. Furthermore, PROS utilizes hash tables to facilitate the retrieval of adjacent road segments.

As far as the privacy profile is concerned, in traditional cloaking algorithms for the Euclidean space, a privacy profile is a pair of (k, A_{min}), where k is the number of indistinguishable users and A_{min} is the minimal area required for cloaked regions. However, because PROS is road network based, i.e., in PROS, a cloaked region is in fact a road segment set, we employ (k, L_{min}, N_{min}) as the privacy profile instead, where L_{min} and N_{min} are the minimal total road segment length and the minimal number of road segments required in a cloaked road segment set, respectively. A cloaked road segment set will be expanded iteratively until the privacy profile is satisfied.

Peer-to-Peer Cloaking

As mentioned in the related work section, with the centralized cloaking method, the location anonymizer will very likely become a performance bottleneck when the number of mobile users increases or the locations of mobile users change very frequently. To avoid the

bottleneck problem and a single point of failure, PROS forms a cloaked road segment set by single-hop or multi-hop peer searching, where mobile users communicate with each other to discover nearby peers. In other words, PROS blurs the exact location of a query initiator without any centralized location anonymizer.

With PROS, before launching a query, a query initiator q will broadcast a probe message, which includes the hop count h specified by q to its neighboring peers and then q listens to the network to wait for neighboring peers' replies. When a peer receives a probe with greater than 1 h value, it not only responds to the probe but also modifies the probe message by decreasing h by 1 and then re-broadcasts the probe. On the contrary, when a peer receives a probe with the h value equal to 1, it only answers to the probe. Consequently, the probe will flood locally within hop count h in wireless networks. After collecting all the replies, the query initiator randomly chooses *k-1* peers to form the cloaked road segment set. And then the Road Segments Set will be expanded randomly to satisfy the privacy parameters L_{min} and N_{min} (therefore, the requestor will not always be in the center of the cloaking result). The cloaked road segment set should (1) include the *k-1* discovered peers and (2) satisfy the requirements of L_{min} and N_{min}. If such a cloaked road segment set cannot be generated, the query initiator has to increase the initial hop count value.

5.4.2 Searching Phrase

As the result of peer-to-peer cloaking, the cloaked road segment set is submitted with the LBS query to the location-based service provider for evaluation. In PROS, we adopt the privacy protected spatial network query algorithms, PSNN and PSRQ, as proposed in [2], for answering K nearest neighbor queries and range queries on road networks. Notice that since the cloaked regions discussed in [2] are based on grid cells rather than road segment sets, we extend both PSNN and PSRQ to retrieve inclusive query result sets based on the input cloaked road segment set and the underlying road network. Finally, the query

initiator filters out the exact query results from the inclusive query result set returned by the Location-based Service Provider.

5.4.3 Demonstration

Cloaked Road Segment Set

Figure 5.6 demonstrates a screen shot of the cloaking module GUI. A user firstly specifies its privacy profile (k, L_{min}, N_{min}) and the hop count h. As the result of cloaking, the blue dot, the green dots, and the yellow dots on the map correspond to the query initiator, the discovered peers, and uninvolved peers, respectively. In addition, the cloaked road segment set is highlighted in red on the map.

Inclusive and Exact Result Sets

The searching module GUI is illustrated in Figure 5.7. Users can specify query types (K nearest neighbor query, range query, etc.). Then, users select the POI types to be searched. Taking the cloaked road segment set as the input, the location-based service provider will return an inclusive query result after the search. The GUI shows an example of a 3-nearest neighbor query. The squares on the map represent the inclusive result set answered by the service providers and the green squares are the exact results filtered out by the query initiator.

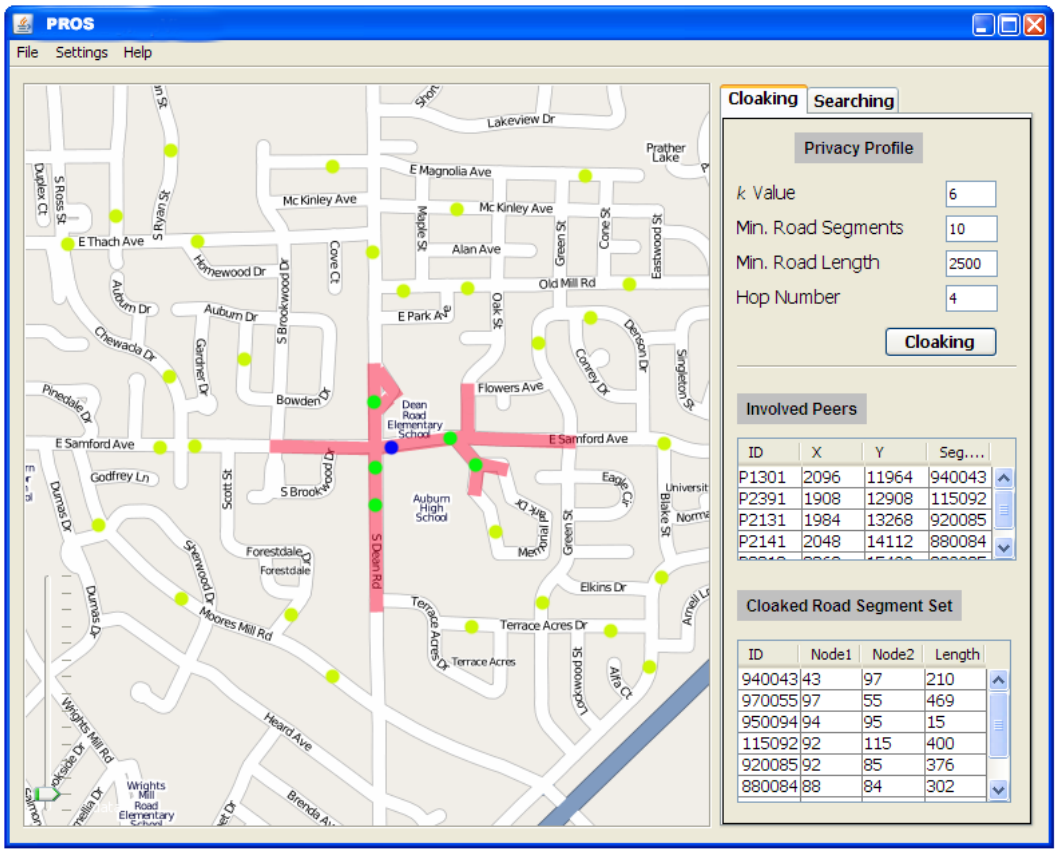


Figure 5.6: Cloaked road segment set.

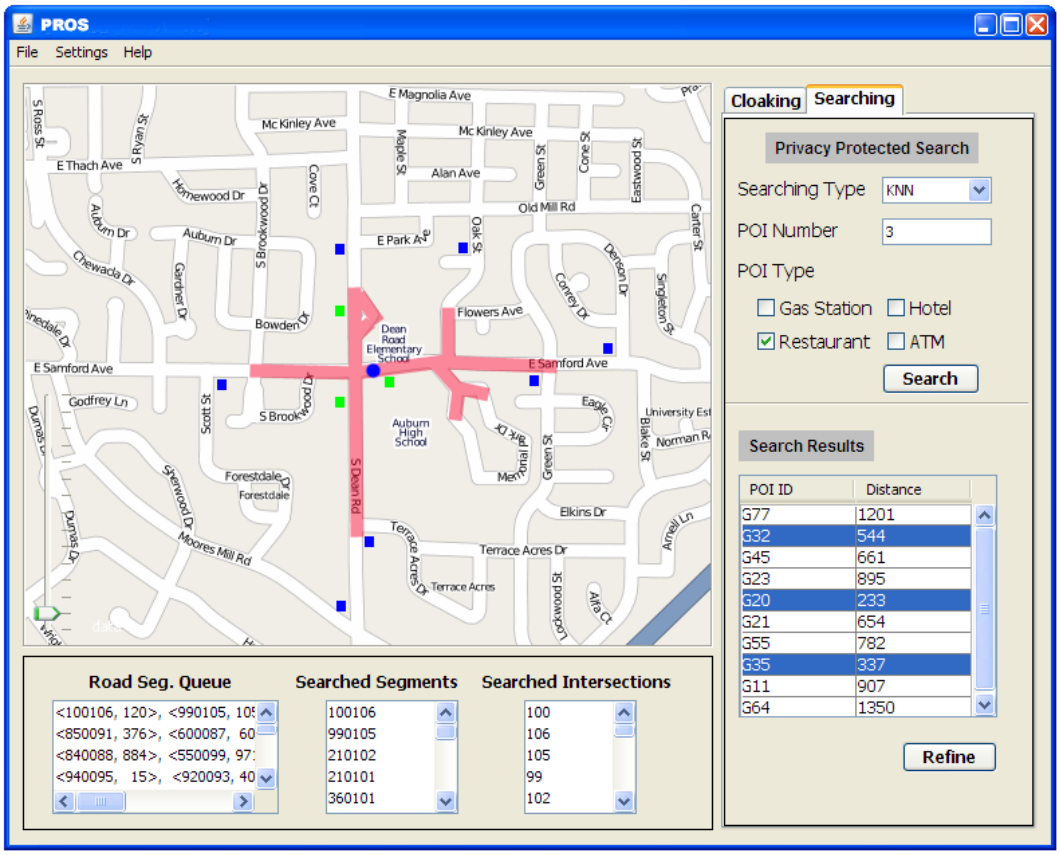


Figure 5.7: Inclusive and exact query results.

Chapter 6

Conclusion and Future Work

In this thesis, our contribution can be summarized as follows:

We introduced the basic ideas of location based services, while raised the potential privacy issues in such kind of the services. And we also surveyed some existing and popular privacy protection solutions in location based services.

A novel cache management framework is proposed which can be applied in the location anonymizer. Our solution can further improve user privacy protection, save computational resources, and decrease communication costs significantly. The experiment results has proofed that our method can increase cache hit ratio remarkably. And for the future extension of this research work, we plan to propose solutions to support more spatial query types and also experiment the corresponding performance of our mechanisms.

Moreover, we proposed a road network based decentralized spatial cloaking method as well as the corresponding incremental spatial query processing algorithms. And we also first raised the issue of indistinguishability for the K-anonymity idea. By using a incremental cloaking region construction process, we are able to provide a higher privacy protection for the query issuer. And we extend the framework to adapt with the road networks to further improve the accuracy of the query results.

But there are still some open issues we will continue to work and improve, for example, the road network indexing method and the peer management schemas. We hope to address these issues in the future researches and propose some more efficient solutions.

Bibliography

[1] Claudio Bettini, Xiaoyang Sean Wang, Sushil Jajodia. Protecting Privacy Against Location-Based Personal Identification. In *Security Data Management*, pages 185-199, 2005.

[2] Thomas Brinkhoff. A Framework for Generating Network-Based Moving Objects. *GeoInformatica*, 6(2):153-180, 2002.

[3] Chi-Yin Chow, Mohamed F. Mokbel. Enabling Private Continuous Queries for Revealed User Locations. In *International Symposium on Spatial and Temporal Databases (SSTD)*, pages 258-275, 2007.

[4] Chi-Yin Chow, Mohamed F. Mokbel, and Xuan Liu. A Peer-to-peer Spatial Cloaking Algorithm for Anonymous Location-based Service. In *Proceedings of the 14th ACM International Symposium on Geographic Information Systems*, pages 171-178, 2006.

[5] Alon Efrat and Arnon Amir. Buddy Tracking - Efficient Proximity Detection Among Mobile Friends. In *IEEE International Conference on Computer Communications* (INFOCOM), 2004.

[6] Bugra Gedik and Ling Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proceedings of the 25th International Conference on Distributed Computing Systems*, pages 620-629, 2005.

[7] Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. PRIVE: Anonymous Location-based Queries in Distributed Mobile Systems. In *Proceedings of the 16th International Conference on World Wide Web* (WWW), pages 371-380, 2007.

[8] Marco Gruteser and Dirk Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the First International Conference on Mobile Systems, Applications, and Services* (MobiSys), 2003.

[9] Haibo Hu, Jianliang Xu, Wing Sing Wong, Baihua Zheng, Dik Lun Lee, Wang-Chien Lee. Proactive Caching for Spatial Queries in Mobile Environments. In *Proceedings of the 21st International Conference on Data Engineering* (ICDE), pages 403-414, 2005.

[10] Wei-Shinn Ku, Roger Zimmermann, Wen-Chih Peng, and Sushama Shroff. Privacy Protected Query Processing on Spatial Networks. In *Proceedings of the 23rd International Conference on Data Engineering Workshops*, pages 215-220, 2007.

[11] Wei-Shinn Ku, Roger Zimmermann, Haixun Wang. Location-based Spatial Queries with Data Sharing in Wireless Broadcast Environments. In *Proceedings of the 23rd International Conference on Data Engineering* (ICDE), pages 1355-1359, 2007.

[12] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases* (VLDB), pages 763-774, 2006.

[13] Kyriakos Mouratidis, Panos Kalnis, Gabriel Ghinita and Dimitris Papadias. Preventing Location-Based Identity Inference in Anonymous Spatial Queries. *IEEE Trans. Knowl. Data Eng.*, 2007.

[14] Sarah Spiekermann. General Aspects of Location Based Services. *Location-Based Services*, pages 9-26. 2004.

[15] Latanya Sweene. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557-570, 2002.

[16] Stefan Steiniger, Moritz Neun and Alistair Edwardes. Foundations of Location Based Services, technical report, University of Zurich, 2006.

[17] Yu Chen, Jie Bao , Wei-Shinn Ku, and Jiun-Long Huang, Cache Management Techniques for Privacy Protected Location-based Services, *Proceedings of the 2nd International Workshop on Privacy-Aware Location-based Mobile Services (PALMS), in conjunction with the 9th International Conference on Mobile Data Management (MDM 2008)*, 2008.

[18] Shu Wang, Jungwon Min and Byung K. Yi, Location Based Services for Mobiles: Technologies and Standards, *IEEE International Conference on Communication (ICC)*, 2008.

[19] Location Based Services, http://en.wikipedia.org/wiki/Location-based_services.

[20] Gruteser, M., Grunwald, D., Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking, *MobiSys* pp. 31-42, 2003.

[21] Du, J., Xu, J., Tang X. and Hu, H., iPDA: Support- ing Privacy-Preserving Location-Based Mobile Services, *MDM* pp. 212-214, 2007.

[22] Duckham, M. and Kulik, L., A Formal Model of Obfuscation and Negotiation for Location Privacy, *Pervasive* pp. 152-170, 2005.

[23] Kido, H., Yanagisawa, Y. and Satoh, T., An Anony- mous Communication Technique using Dummies for Location-based Services, *IEEE International Conference on Pervasive Services* pp. 1248, 2005.

[24] Ghinita, G., Kalnis, P. and Skiadopoulos, S., MobiHide: A Mobilea Peer-to-Peer System for Anonymous Location-Based Queries, *SSTD* pp. 221-238, 2007.

[25] Po-Yi Li, Wen-Chih Peng, Tsung-Wei Wang, Wei-Shinn Ku, and Jianliang Xu, A Cloaking Algorithm Based on Spatial Networks for Location Privacy, *In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, 2008.

[26] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Shenker, A Scalable Content-Addressable Network, *In Proceedings of ACM SIGCOMM* 2001

[27] Foxs News: Man Accused of Stalking Ex-Girlfriend With GPS. http://www.foxnews.com/story/0,2933,131487,00.html. Sep 4, 2004

[28] USAToday: Authorities: GPS System Used to Stalk Woman. htt://usatoday.com/tech/news/2002-12-30-gps-stalker x.htm. Dec 30, 2002

[29] Voelcker, J.: Stalked by Satellite: An Alarming Rise in GPS-enabled Harassment. *IEEE Spectrum 47(7) 15-16* (2006)