An Approach for Ensuring Robust Support for Location Privacy and Identity Inference Protection

Chowdhury Sharif Hasan
Marquette University

AN APPROACH FOR ENSURING ROBUST SUPPORT FOR LOCATION PRIVACY AND IDENTITY INFERENCE PROTECTION

by

Chowdhury Sharif Hasan

A Thesis submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

August 2010

ABSTRACT

The challenge of preserving a user's location privacy is more important now than ever before with the proliferation of handheld devices and the pervasive use of location based services. To protect location privacy, we must ensure k-anonymity so that the user remains indistinguishable among *k-1* other users. There is no better way but to use a location anonymizer (LA) to achieve *k*-anonymity. However, its knowledge of each user's current location makes it susceptible to be a single-point-of-failure. In this thesis, we propose a formal location privacy framework, termed SafeGrid that can work with or without an LA. In SafeGrid, LA is designed in such a way that it is no longer a single point of failure. In addition, it is resistant to known attacks and most significantly, the cloaking algorithm it employs meets reciprocity condition. Simulation results exhibit its better performance in query processing and cloaking region calculation compared with existing solutions. In this thesis, we also show that satisfying *k*-anonymity is not enough in preserving privacy. Especially in an environment where a group of colluded service providers collaborate with each other, a user's privacy can be compromised through identity inference attacks. We present a detailed analysis of such attacks on privacy and propose a novel and powerful privacy definition called *s*-proximity. In addition to building a formal definition for *s*-proximity, we show that it is practical and it can be incorporated efficiently into existing systems to make them secure.

ACKNOWLEDGMENTS

Chowdhury Sharif Hasan

The thesis was made possible due to the continuous support and contributions from my Advisor, Thesis committee members, fellow Lab mates and my family members. First and foremost, I am thankful to Dr. Sheikh Iqbal Ahamed, Director, Ubicomp Lab, for not only did he remain my academic advisor throughout the course of study, but he was also the continuous source of inspiration and encouragement for my research endeavors. I am also grateful to the thesis committee members Dr. Douglas Harris, Dr. Praveen Madiraju and Dr. Rong Ge for their invaluable comments and patience during the preparation of the thesis. I would like to express my gratitude for my fellow Ubicomp lab members for their benevolent support and valuable advice. I feel highly honored and would like to thank the Dept of MSCS for providing me the opportunity to work as a TA/RA throughout my study period as a Master's student.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ELABORATIONS

Term	**Elaboration**
LBS	Location based Service
LBQ	Location based Query
POI	Point of Interests
LA	Location Anonymizer
CR	Cloaking Region
ASR	Anonymized Spatial Region
NNC	Nearest Neighborhood Cloaking
IC	Interval Cloak
Casper	New Casper Cloak
HC	Hilbert Cloak
QoS	Quality of Service
CP	Cloaking Primitive
POLS	Privacy Observant Location Service
PDA	Personal Digital Assistant
PND	Personal Navigation Device

Chapter 1: Introduction

The use of Location Based Services (LBS) has become ubiquitous with the growth of handheld devices, PDAs, smart phones and GPS enabled cars. LBSs will flourish even more with the surge of new genre of information systems. While requesting for a location based service, a user can easily mask his identity using un-linkable pseudonyms [Hoh05] but he needs to provide his location information, even if with less precision. The location service provider (LSP) or an adversary, who secretly listens to the communication channel between a user and the LSP, builds his own chronological record of location data over the period of time. Later this knowledge might be used by him with techniques like correlation attack, restricted space identification [Gruteser03], observation identification [Gruteser03, Mokbel06] etc to link the records with actual user identity. The user's personal preference, state of health, political view etc can be even inferred from the places he visits or visited. No wonder, his mail boxes may be inundated with unwanted advertisements. This is location privacy violation and due to its possible aftermaths preservation of an individual's location privacy is of utmost importance.

There are several ways to thwart location privacy violation. One obvious solution is to obfuscate, i.e. deliberate degradation of geographic position [Duckham05], the location of the query issuer but it degrades the quality of service. Most importantly, the query requester cannot be sure of meeting his *k*-anonymity [Hoh05] (i.e. in his obfuscated region there are at least *k-1* other users) requirement in this process as he does not know the actual position of other users. Achieving *k*-anonymity in location privacy requires that the probability of an attacker re-identifying a user from the obfuscated region does not exceed 1/k [Monjur09]. Indeed, the user can communicate with his peer devices, establish trust, share location among them and thereby construct a region where his *k*-anonymity is satisfied [Chow06, Hashem07]. However, in pervasive environment, establishing trust takes time and due to mobility of the devices such peer group has short-lived existence [Satya96] which makes peer-based systems infeasible. Yet another and most widely accepted way is to use Location Anonymizer (LA) [Gruteser03, Gedik05, Gedik08, Ghinita07, Bamba08, Kalnis07] which sits between users and LSPs. All subscribed users provide their exact locations to the LA periodically. This enables LA to construct a bounded region, known as cloaked region (CR), for each user, satisfying their *k*-anonymity and other privacy requirements. LA is, however, not the ultimate solution because:

1) LA is a bottleneck in communication.

2) Each user has to give his exact location and for that LA is attractive to adversaries (possibly a Big Brother) as compromising it means gaining all the user location data.

3) The user needs to secure the communication channel with LA which is costly.

4) Some cloaking techniques are very costly in both computation by LA and query processing by LSP.

*5) There are well known passive attacks against most of the cloaking techniques [*Kalnis07, *Gruteser03] and*

6) All of the existing cloaking techniques (except [Ghinita07] & [Talukder10]) fail to meet reciprocity condition [Kalnis07].

The reciprocity condition necessitates that every user in an anonymization set (AS) also generates the same AS for the given anonymity requirement (i.e. *k*) [Kalnis07].

In this thesis, we present an efficient location privacy safeguard, titled SafeGrid that makes best use of both obfuscation and cloaking techniques and it can be used with or without an LA depending on the user's choice. In SafeGrid, we divide the entire space into obfuscation cells and these cells are built from all the users' obfuscation requirements. Instead of giving his location to LA the user provides one or more obfuscation cells. In fact, SafeGrid negotiates with the user the degree of obfuscation required in the event a location anonymizer is compromised. Obfuscated cells are negotiated earlier and updated after a considerable period of time. LA in SafeGrid uses a cloaking technique that strictly meets reciprocity condition [Ghinita07, Kalnis07]. In this approach, all LBSs have prior knowledge of the obfuscation cells so that they can pre-process POIs for each cell and can readily provide results to LA or the user.

Although SafeGrid provides performance gain and enhanced attack resistance, the cloaking algorithm is based on satisfying *k*-anonymity. However, in this thesis we argue that *k*-anonymity does not provide sufficient protection against privacy violation. We present two attacks, the *heterogeneity attack* and the *conformity attack*, and we show how they can be used to compromise a *k*-anonymous location based query. The heterogeneity attack reveals that *k*-anonymity can create groups that fail to provide overall anonymity due to lack of sufficient match among the members with respect to some sensitive user attribute. Likewise, approaches satisfying only *k*-anonymity disregard consequences of revealing important context [Talukder08] information though the service request and pave the way for conformity attack. Besides illustrating the attacks with real world examples, we have provided their formal definitions which clarify how they relate to the contexts [Talukder08] of the query and static information [Machanavajjhala06] of the users in the anonymity set (AS) [Machanavajjhala06]. Most of the existing approaches [Mokbel06, Gruteser03, Kalnis07, Ghinita07, Gedik05, Duckham05, Ghinita08] are vulnerable to these attacks as they undergo following problems.

1) *They choose k number of users for constructing a CR on the basis of current locations of the users only ignoring any other relevant static attribute of the users being grouped together.*

2) *Most of the approaches forward the query to the LSP without making any modification. However, a query that contains request for a specialized service may disclose a number of contexts and static user information on those contexts hastens re-identification.*

3) *They do not consider preference of users regarding any other contexts of their interest beyond location on the basis of which they want anonymization.*

As an attempt to guard against the above mentioned attacks we have introduced a new notion of privacy, called *s-proximity*, which requires that each anonymity set (AS) contains at least s members belonging to the *equivalence class* of the query issuer. An *equivalence class* is defined to consist of users having high correlation with the actual query requester with respect to a set of static user attributes [Machanavajjhala06]. With this new privacy parameter a user's privacy profile [Mokbel06] takes the form of $< k, s, A_{min} >$. We propose a pragmatic solution that offers services with such privacy protection. Our approach, i.e. the SafeGrid framework, uses a trusted third party to mediate user's query to the LSP. In order to incorporate the s-proximity measure, we consider the trusted third party to be equipped with more capabilities and extended functionalities. We call this trusted third party *Context Aware Location Anonymizer (c-*

LA) as it is featured with additional modules for *context based query generalization*, *proximity group formation* as pre-steps of CR generation. We propose a novel algorithm called *Selective Nearest Neighbor (SNN)* for AS construction and CR generation. A formal proof establishes that SNN provides enhanced privacy by reducing the probability of re-identifying the actual query issuer. Implementation of the system validates the feasibility of our proposed approach.

In summary, the contribution of this thesis lies in designing a LBS framework which improves both performance gain and attack resistance. The proposed framework, SafeGrid never uses the user's exact location; neither in communication nor in computation. LA in it has none of the above mentioned problems. Evaluation shows its better performance in CR construction and query processing with less number of POIs returned to the client than any of the known works in literature. It meets a user's *k*-anonymity every time, if the required *k* is not greater than total number of subscribed users. In addition, whatever may be the required *k*, achieved *k* for each user is close to maximum of all *k*. Most notable feature of our approach is that it generates CRs which meet reciprocity condition. A number of attacks can be formulated if the cloaking technique does not satisfy reciprocity condition. In addition, we have proposed a novel privacy measure called *s*-proximity and shown how our approach meets reciprocity condition along with s-proximity. We have presented several attacks on location privacy and unintentional identity inference with illustrative examples and shown that the cloaking techniques that satisfy reciprocity condition and *s*-proximity can overcome those attacks. We have conducted extensive performance analysis of our proposed model.

Evaluation results demonstrate its feasibility as a practical solution and improvement over existing approaches.

The outline of the rest of this thesis is: Chapter 2 contains background information. The proposed grid based location privacy framework has been described in Chapter 3. The details of identity inference protection and enhancements to the initial framework are provided in Chapter 4. Chapter 5 contains the current state-of-the-art and related work. The comparative analysis and experimental results of our approach are presented in Chapter 6. Finally, our future research direction and concluding remarks can be found in Chapter 7.

Chapter 2: Background

This section discusses definitions and brief descriptions of relevant technical terms used throughout the thesis, in order to facilitate understanding of the materials presented.

2.1 Pervasive Computing

Pervasive computing is the notion of making computing services available anytime, anywhere and on demand basis to the user [Robinson04]. Starting its journey as “Ubiquitous computing” at Mark Weiser’s [Weiser91] Xerox PARC lab (through his seminal paper in 1991), it has emerged as “the computing for the 21st century”. This technology is especially a synergy of diversified concepts such as mobile computing, wireless network, embedded computing, context-awareness with sensor technology and human computer interaction. The pervasive computing environment comprises devices of heterogeneous platforms and capability. Despite the advancement of handheld device technology (e.g. PDAs, smart phones, etc.) in recent years, these devices are suffering from a number of challenges to date [Satya96, Want05], which include but are not limited to inadequate processing capability, restricted battery life, limited memory space, frequent disconnection, and limited bandwidth. The applications developed for this environment emphasize performance, and the efficient and stingy usage of resources in the devices.

2.2 Context

We reiterate the formal definition of Context from A. Dey: “Context is any information that can be used to characterize the situation of the entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves” [Dey01]. Pervasive computing environment is intended to facilitate humans to convey situational information or contexts through widening the conversational bandwidth. This richness of communication sets apart the context-aware applications from the traditional applications. Location is the most common form of context found in Location based Applications and is readily available through global positioning systems (GPS). Distributed systems are using the location information available from underlying communication infrastructure, for instance cell information in cellular networks such as GSM [Schmidt99]. Contexts may take the form of human factors such as personal information about the user (knowledge of habits, emotional state, bio-physiological conditions), the user’s social environment (co-location of others, social interaction, group dynamics), and the user’s tasks (spontaneous activity, engaged tasks, general goals). Likewise, context related to the physical environment can be location (absolute position, relative position, co-location), infrastructure (surrounding resources for computation, communication, task performance), and physical conditions (noise, light, pressure) [Schmidt99].

2.3 Location based Service

Location based Application or Location based Service (LBS) is a special form of context-aware application, where only location information is used as contextual information and included inside the query. Some of the interesting applications are worth discussing before we delve into their privacy concerns. Here are some of them:

Tracking service: This service allows a user to locate another person when some time and location constraints are met. For example, in order to offer quality service to their customers, employers may need to track employees' whereabouts and modify the nearest employee's work schedule to meet customer needs. Parents may want to track their children through cell phones while their children are roaming in a particular area. FindYourChild [FindChild] provides such services to its cellular phone subscribers. And more recently, PocketFinder Application [PocketFinder], downloadable through G1 phones preloaded with Google Android, provides the same service in US.

Buddy Finder: This is a friend finder application that runs on mobile handheld devices and allows groups of friends to show one another exactly where they are and what they're doing. The example of such an application is Buddy Beacon [BuddyBeacon], which is available for most mobile phones including iPhone and allows the users to connect with their friends even across different carriers. The difference between the tracking service and Buddy Finder service is that the location update information is sent to friends only with the user's approval.

What's Here?[Gunter04]: This is essentially a local search service for events and places in a locality. Examples include a list of forthcoming events in a particular building, tourist points of interest, or the route to the nearest restaurants. The location queries are made through an Anonymizer in order to hide the user's exact location information from the Location based Server. iPhone application Earth Comber [Earth Comber] is able to find and notify the user on the closest free WiFi hotspots, coffee joints, pubs, eateries, fuel, stores, ATMs, live music, movies by title and much more.

Calendar service, bulletin board service (Constraint based on presence of the owner of the resource or the requester of the service): Data from these services will be shared with the requester of the service based on some location constraints being met by the request issuer [Hengartner06]. For example, the requester may have to reside inside the office of the owner of the information in order to get access to the information. The owner of the resource is in charge of setting the time and location preferences for accessing the information.

Personal navigator (Campus Locator Service): The service can help a person find out locations around the campus through a central location server or by accessing a local database search, sensing the nearby access points. The latter technique requires the mobile device to keep a local database that maps sensed access points to campus locations. To learn more details about the technique, please refer to the Related Work chapter for Privacy Observant Location Service [POLS].

*Similar-interests service***:** This service gathers data from nearby cellphone users and notifies other cellphone users within the vicinity that have similar interests.

Coffee Shop, Restaurant, Cinema hall, Stadium seat availability: Another class of applications known as opportunistic sensing application [Kapadia08] exploits the idea of sharing contextual information among users in the form of periodic reports and thus eliminates the necessity of mounting all the sensing equipment on every single device.

The anonymized approach depicted in Hitchhiking [Tang06] is used to infer coffee shop space availability. Without much extension, the idea can be used to find availability in places like restaurants, cinema halls, and stadium seating. AnonySense [Kapadia08], the architecture for opportunistic sensing of contexts, deploys a third party task server and application server to blur the location information and performs aggregation of reports to preserve privacy.

Shopping Mall Crowds Pattern: Similarly, the above approach can be used to identify which shops are most popular, based on the presence of people.

Traffic Pattern Monitoring System: This service helps a driver to know the traffic pattern of the route or zone to which he is headed. He can plan accordingly to reroute his travel, depending on the congestion information provided. Dash™ and TomTom® are examples of traffic navigation systems where user locations are transmitted to servers, and the routes with optimal distance and least traffic congestion are presented.

Market Model [Gunter04]: This is a type of service where a person can participate in an anonymous survey providing personal information to help build market characteristics for a group of users. The group may satisfy a certain time/space criterion. An example could be the age pattern or average income of commuters at certain time periods at a station. This service may serve the marketing needs of a company operating business in a specific area.

2.4 Privacy

Westin's definition of information Privacy states "Privacy is the claim of individuals, groups or institutions to determine for themselves when, how and to what extent information is communicated to others" [Westin67]. The privacy violation in pervasive computing is thought to be the most contentious of the issues addressed so far. With the multitude of information flowing around the environment, individual privacy is now an easy prey for the eavesdropper, stalkers and adversaries. Researchers have made continual efforts to define privacy that best suits the pervasive computing environment. Palen and Dourish [Palen03] defined privacy in socio-technological terms for the networked world and described it as the continuous balance of disclosure, identity and temporal boundaries. Classical Design guidelines for Ubiquitous Systems by Langheinrich [Langheinrich02] present six principles that are common in most legislative systems. He describes the social implications of the ubiquitous system and claims that the guidelines are readily implementable in the environment. Emphasis is on choice and consent, pseudonymity, locality, and security measures which are basic design needs for every ubiquitous computing application. To summarize the privacy implication of today's digitized world, we reiterate the famous quote of Ron Rivest: "What was once private is now public", "what was once hard to copy is now trivial to duplicate" and "what was once easily forgotten, is now stored forever". So, information transcends boundaries of space and time.

2.5 Privacy issues in Location based Service

The LBS queries such as "find the nearest shopping mall" or "find the nearest Italian restaurant" may seem innocuous, but once the identity of the individual making the query is revealed, in the near future she might be bombarded with unsolicited advertisements, newsletters, etc. So, in naïve terms, protecting privacy in LBS means in some way hiding the exact location of the user making a query. As briefly discussed earlier, the task of the Location Anonymizer is to obfuscate or cloak the location information of the query issuer with several other persons in the vicinity. For an anonymity level *k* as requested by the query issuer, the LA will generalize the request (for LBS) in such a way that it will contain the location information indistinguishable from *k*-1 other users. The principle is known as *K*-anonymity [Sweeney02] and has been introduced for privacy protection of microdata such as census figures, medical information, and voting registration. In LBS the concept can be adapted as: a *K*-anonymized query request ensures that the CR chosen for a query offers the attacker a probability of re-identification not exceeding 1/*K*, *K* being the preferred anonymity level (in short anon level) of a query issuer. Details of the privacy protection mechanisms can be found in the Related Work chapter. Listed here are some of the useful terms related to privacy protection in LBS used throughout the thesis.

2.5.1 Service Provider

Service Providers (SP) are the entities that share or provide a requested service/resource. In the case of LBS, the SP provides the result of LBQ to the Servicer Requester. The LBS server is considered to be a non-trusted entity. The assumption is that the commercial service can collect unauthorized information about individuals and use it later for advertisement and marketing purposes. The communication channel between the LBS and LA is not considered secure.

2.5.2 Service Requester

Service Requesters (SR) are devices that request a specific service/resource. As discussed earlier, the devices are constrained with limited battery power and network bandwidth. The Service Requester issues the query through a Location Anonymizer (LA) to ensure the user's required level of privacy (level of anonymity). The LA obfuscates the exact location of the request issuer and forwards the query to LBS.

2.5.3 Location Anonymizer

The Location Anonymizer (LA) is a trusted server, whose task is to collect the current location information of the users (while making the query) and anonymize their queries. Associated with each query is a required degree of anonymity *k*, which ranges between 1 (no privacy requirements) and the user cardinality inside the anon set (maximum privacy) [Kalnis07].

2.5.4 Cloaking Region

The location information forwarded by LA to LBS is termed as the Cloaking Region (CR) or Anonymized Spatial Region (ASR). CR contains *k* users within it, *k* being the desired anonymity level of the request issuer. The choice of the CR varies greatly depending on different cloaking algorithms. The evaluation section provides a comparative analysis of different cloaking algorithms including our approach.

2.5.5 Anonymity set

The anonymity set is defined as the list of users within the same CR. The location represented by a CR is the same for all members of the anonymity set.

Chapter 3: The SafeGrid Location Privacy Framework

In this section, we introduce our location privacy framework, SafeGrid by providing a brief overview. Our approach is based on grid structure. We are using a novel type of grid structure which is simple and efficient. We have included an additional component for performing the task of grid management. The Location Anonymizer (LA) included in our framework is used based on user's decision i.e. SafeGrid offers its subscribers the alternatives of directly submitting query to LSP or going through an intermediate LA. In SafeGrid, whether an LA is used or not, an initial grid is built. All users, location service provider (LSP) and LA have knowledge of this grid. It is fundamental to understand the design of the grid we will be using. So, here we are discussing the design issues related to our proposed grid structure.

3.1 Grid Structure

We consider entire region of service to be composed of cellular units in the form of a grid structure. The smallest unit of the grid is called a *cell*. For the sake of simplicity, a cell is assumed to be a square-shaped geographical block. Each cell, denoted by a global cell no ($Cell\_ID$), has an area ($Cell\_Size$) which is set by fixing the length of one side of the square shaped cell. We term this dimension as $GridRes$ which is an important parameter. A user's location identity is represented by the cell he is in. A user is never required to reveal his exact location rather he provides his cell identity. Hence, users may directly contact an LSP to place his request or may go through a third-party Location Anonymizer. Although it may seem unnecessary to have an LA, later we mention the obvious advantages of using it. Thus we present a framework which has a Location

Anonymizer as an optional component. Users are not obligated to get through an LA though they have option to make use of it for gaining more privacy protection.

3.2 Grid Manager (GM)

We incorporate this auxiliary component for handling the management of location grid. It has the relevant information like width and height of the area of concern. GM considers lower left corner of that area as the point of origin of the grid (POG). Most important part of grid design is to fix the size of its cells. $Cell\_size$ is an important parameter, as it has various impacts on the whole system. Primary level of anonymity and quality of service at client side are directly related to this parameter. Choosing an optimal value for $Cell\_size$ is a difficult task. Moreover, its optimality also varies from user to user because users with varying processing capability and diverse privacy requirements may possess individual preference regarding cell-size. That's why it is important to consider all subscribed users' preference while choosing a global cell-size for the entire grid. In our proposed approach Grid Manager chooses $Cell\_Size$ according to the following algorithm.

GRID-RESOLUTION-COMPUTE (*U*)

```
$N \leftarrow length[U]$
for $i \leftarrow 1$ to $N$
    do $a_{min} \leftarrow a_{min} + U[i].min\_a$
       $g_{min} \leftarrow g_{min} + U[i].min\_g$
$a_{min} \leftarrow a_{min}/N$
$g_{min} \leftarrow g_{min}/N$
for $i \leftarrow 1$ to $N$
    do $\delta_a$ += $(U[i].min\_a - amin)$ ^ 2
$A_{min} \leftarrow \infty, A \leftarrow Nil$
```

```
10. for $i \leftarrow 1$ to $N$
11.     do if $a_{min} \leq (U[i].min\_a + 2\delta_a)$
12.         then $A \leftarrow U$[i]
13.             if $U[i].min\_a < A_{min}$
14.                 then $A_{min} \leftarrow U[i].min\_a$
15. $G_{min} \leftarrow 0$, $G \leftarrow Nil$
16. for $i \leftarrow 1$ to $N$
17.     do if $U[i].min\_g \leq A_{min}$
18.         then $G \leftarrow U[i]$
19.             if $U[i].min\_g \geq G_{min}$
20.                 then $G_{min} \leftarrow U[i].min\_g$
21. $Cell\_Res \leftarrow G_{min}$
22. return $Cell\_Res$ and $TS$
```

Each subscribed user has a couple of privacy preference parameters a_{min} and g_{min} which define minimum size of cloaked region and minimum cell size respectively. The algorithm starts with each subscribed user sending these parameters to the GM. The **for** loop of lines 2-4 combines the inputs from all the users. In order to eliminate feedbacks from any possible malicious user, inputs that lie at extremities are excluded based on the threshold value δ_a. The **for** loop of lines 10-14 computes the ultimate lower bound of cloaked region, A_{min} after excluding the extreme values. Likewise, the lower bound of cell size, G_{min} is calculated in the **for** loop of lines 16-20. These filtered inputs are considered to fix a $Cell\_Res$ which meets every user's g_{min} requirements and ensures that it is smaller than every user's a_{min} requirement. A user can compare the $Cell\_Res$ returned by GM with his a_{min} and g_{min} parameter to verify any malicious attempt by GM. Due to this, GM does not need to be a trusted entity as it can't maliciously manipulate grid resolution. GM performs this computation periodically with an interval of δ amount of time. At time t GM computes grid resolution with the inputs received from the subscribed users. It stores the time of this computation as TS along with the validity period of this calculated grid resolution (i.e. $Cell\_Res$).

3.3 SafeGrid: Without Location Anonymizer

Our preliminary system, which is built on the proposed grid structure, provides a location privacy solution without any location anonymizer. Figure 3.1 depicts how a user communicates with GM and LSP in such a system.

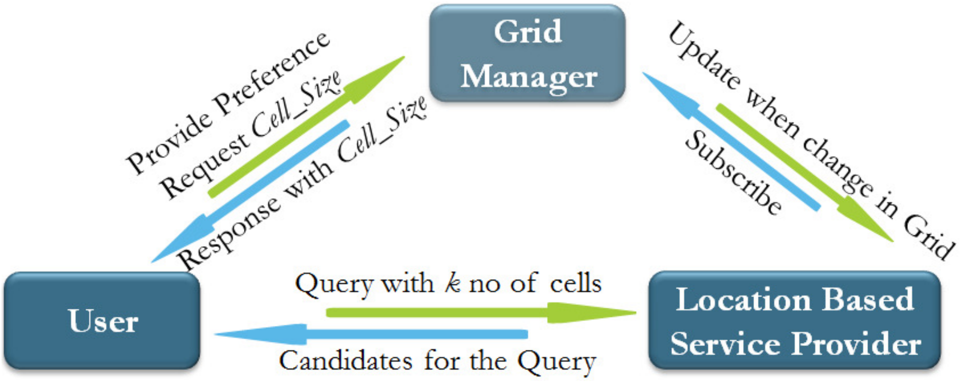


Figure 3.1. Primitive Framework (Without Location Anonymizer)

In the initialization phase, users send their privacy preference parameters like a_{min} and g_{min} to the Grid Manager. Upon receiving data from all users GM fixes the grid resolution using the algorithm discussed earlier. Then it informs users and LSP regarding this change in grid resolution along with the duration of its validity. User submits his location based queries directly to LSP using his knowledge of grid structure. Despite sending his own location, user includes a set of k cells in his query. LSP responds with the list of point of interests (POI) for all those k no of cells. Although query results are returned to the user for *k* no of cells, user considers only the ones related to his own cell ignoring the rest. As this system eliminates location anonymizer, it provides a number of obvious advantages. *First*, query processing time is reduced significantly. *Second*, user's exact location is never revealed to any third party. *Finally*, it eliminates the requirement of secure communication channel between communicating entities. Taking these issues into consideration this sort of location privacy framework seems suitable for pervasive computing environment.

So far, we have explained our basic approach which is a naïve grid based system without LA. Now we discuss that this solution is not enough for securing location privacy and a third party LA is indeed required in such a framework. We continue our step by step discussion towards our eventual SafeGrid framework which is reliable and efficient in terms of privacy preservation and quality of service.

3.4 SafeGrid: Grid with Location Anonymizer

While making a location based query, a user's location is obfuscated among at least *k* cells, including his own cell. This approach does not guarantee that the generated Anonymization Set (AS) [Gruteser03] contains at least *k* no of users. In order to ensure that user's k-anonymity requirement is satisfied an LA is incorporated in our framework. Below we present the architecture of the complete system including a location anonymizer.

Now, we present our grid based location privacy framework, SafeGrid. Major components of SafeGrid are GM and LA. A user provides his choice to GM. GM constructs the grid for a certain period of time and sends grid parameters

$< POG, Cell\_Res, \delta >$ to users, LBS and LA. From these parameters every party has information about the entire grid and anyone can compute a $Cell\_ID$. This knowledge of global $Cell\_ID$ for every party is basic to our approach. We discussed about the characteristics of GM in Section II. The architecture of SafeGrid is illustrated in Figure 3.2.

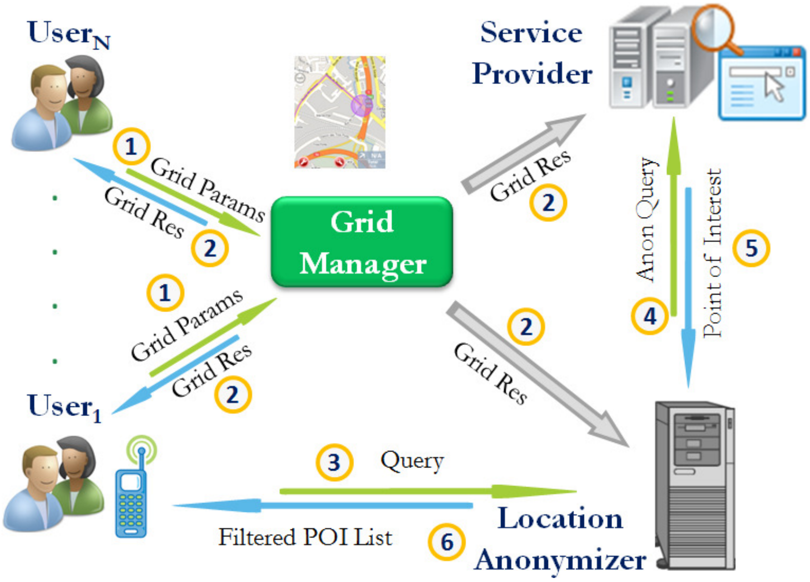


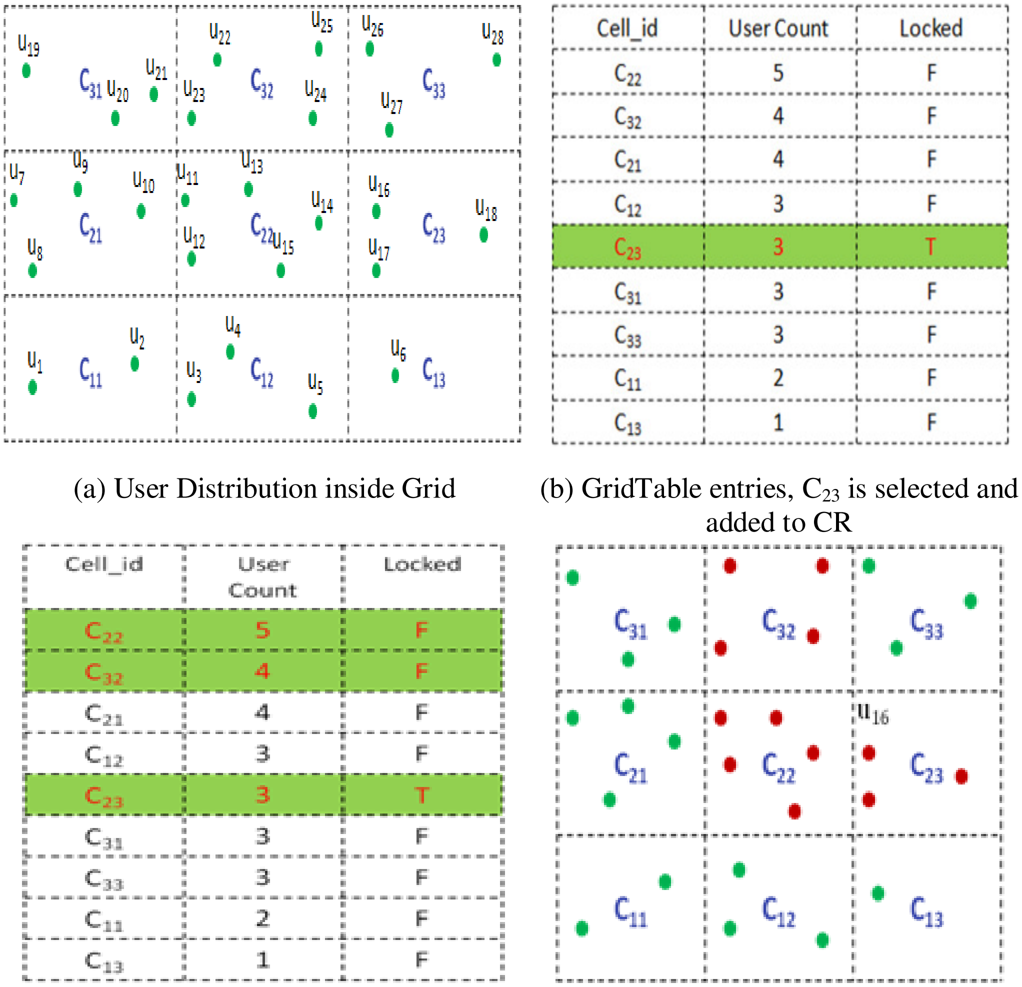
Figure 3.2. Architecture of SafeGrid

Basic purpose of using a Location Anonymizer is to ensure that the query requester is anonymous among at least $(k-1)$ other users. Users send periodic location update information to the LA. With data from all users LA fills up its $GridTable$ which stores the number of users in each cell and a parameter *locked* indicating the availability of the cell in the form of a tuple <*Cell_id, User Count, Locked*>. Then Location Anonymizer uses following algorithm to compute the cloaked region.

BASIC-CR-CONSTRUCT (*GT, U, u_q, k*)

```
Sort $GT$ in descending order of $\eta$
$CR \leftarrow U[q].c$
$CR.count \leftarrow \mathrm{GT}[q].\eta$
$GT[q].locked \leftarrow True$
$j \leftarrow 0$
while $GT$ has more elements
        do if $CR.count \geq k$
              then return $CR$
           else $j \leftarrow j + 1$
                if $GT[j].locked = False$
                   then $CR \leftarrow GT[j].c$
                        $CR.count$ += $GT[j].\eta$
                        $GT[j].locked \leftarrow True$
```

It takes input $< GT, U, u_q, k >$ from the user u_q locating inside cell $u_q.c$. Here k denotes the anonymity requirement of u_q. The ultimate goal of the algorithm is to construct a CR consisting of a number of cells including $u_q.c$ so that the cumulative number of users in those cells is at least k. U is the set of all users along with their current cell location. The procedure initializes cloaked region with the cell of actual query requester. The $GridTable$ managed by LA is sorted according to $User\ Count, \eta$ in descending order. Subsequent cells are taken from this sorted list and added to CR until anonymity level of CR reaches k. LA uses this cloaked region (CR) as the location data to formulate the final query which is sent to the LSP. In the following figures we illustrate how cloaked region is constructed when user u_{16} locating inside C_{23} issues query to LA with anonymity requirement of 10.



(a) User Distribution inside Grid

Cell_id	User Count	Locked
C_{22}	5	F
C_{32}	4	F
C_{21}	4	F
C_{12}	3	F
C_{23}	3	T
C_{31}	3	F
C_{33}	3	F
C_{11}	2	F
C_{13}	1	F

(b) GridTable entries, C_{23} is selected and added to CR

Cell_id	User Count	Locked
C_{22}	5	F
C_{32}	4	F
C_{21}	4	F
C_{12}	3	F
C_{23}	3	T
C_{31}	3	F
C_{33}	3	F
C_{11}	2	F
C_{13}	1	F

(c) Final state of GridTable

(d) CR for user u_{16}; users in the AS are marked as red

Figure 3.3. Steps of the BASIC-CR-CONSTRUCT algorithm

Figure 3.3(a) depicts a sample distribution of users in different grid cells. Subsequent figures show the states of the GridTable maintained by the LA. Finally LA constructs cloaked region for user u_{16} including cells $\{C_{22}, C_{23}, C_{32}\}$ which ensures anonymity of 12. The cloaked region constructed in this way meets the anonymity requirement of the query issuer. However, satisfying k-anonymity does not always guarantee full-proof privacy safeguard. In the next section we present a couple of sophisticated privacy threats which are applicable to most of the existing solutions.

3.5 Location Privacy Attack

This approach, however, reveals some sensitive information to the LSP or any other attacker. We assume that following information is available to the attacker [Ghinita07, Kalnis07, Talukder10]: *1) Knowledge of cloaking algorithm; 2) The cloaked region; 3) Anonymizing set and 4) Required anonymity level of query requester.* If an attacker continuously monitors these information over a period of time, he may infer a user's location. We illustrate a couple of such scenarios below.

Attack Scenario 1: In a grid based approach, CR actually returns a set of grid cells ensuring that those cells contain at least k no of users. Suppose, two such CRs are formed where $CR_1 = \{C_{22,} C_{23}, C_{32}\}$; $AS_1 = \{u_1, u_2,\ u_3,\ u_4,\ u_5,\ u_6\}$ and $CR_2 = \{C_{22,} C_{32}\}$; $AS_2 = \{u_1, u_2,\ u_3,\ u_4,\ u_5\}$. By gaining knowledge of such two CRs and correlating the data a knowledgeable attacker can infer the cell location of user u_6. The scenario is depicted in Figure 3.4.

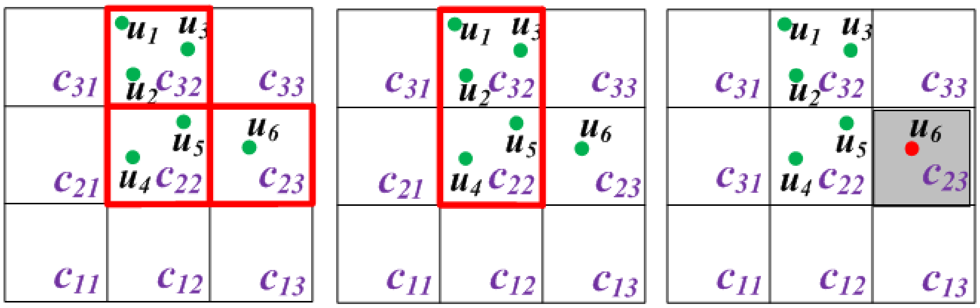


Figure 3.4. Attack scenario 1 for grid based CR computation

All other standard cloaking techniques can be shown to fall under such attack. It can be illustrated through Figure 3.5(a) (b).

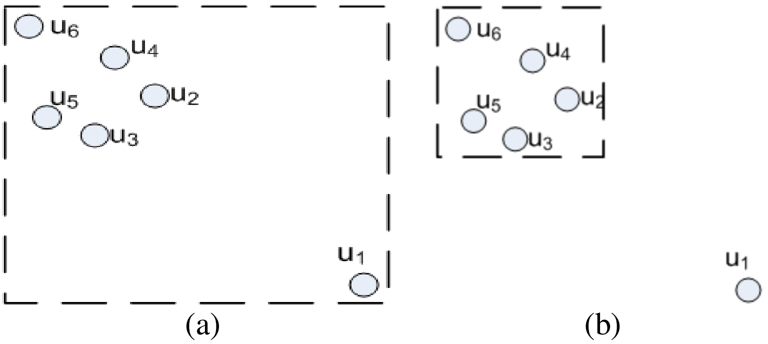


Figure 3.5. Attack scenario 1

Figure 3.5(a) shows that a CR covers 6 users $\{u_1, \ldots, u_6\}$. For instance, user u_1 has issued a query with anonymity requirement $k = 5$. The CR in the figure satisfies the requirement. Later, when any other user with index $2 \le k \le 6$, issues a query with the same anonymity level, the shrunk CR in Figure 3.5(b) is provided to the query processor. According to the assumptions, the attacker may conclude that the last request issuer is the user u_1 and hence his location is at the right corner of the CR.

Attack Scenario 2: Let, a grid based approach constructs cloaking regions as $CR_1 = \{C_{22,}\ C_{23}, C_{32}\}$; $AS_1 = \{u_1, u_2,\ u_3,\ u_4,\ u_5,\ u_6\}$ and $CR_2 = \{C_{13,}\ C_{23}\}$; $AS_2 = \{u_6,\ u_7,\ u_8\}$. By correlating information gained from such two CRs an attacker can infer the cell location of user u_6. Figure 3.6 illustrates this attack scenario.

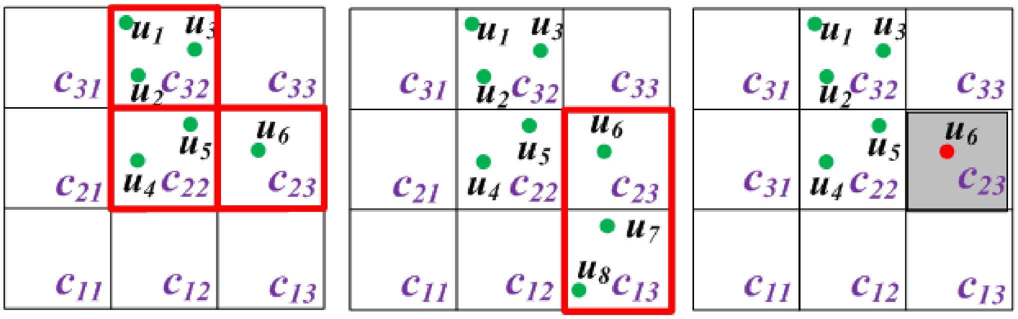


Figure 3.6. Attack scenario 2 for grid based CR computation

Identifying the common cell/cells in the cloaked regions of consecutive queries issued by u_6 , reveals her own cell, i.e. C_{23}. An attacker can easily conceive such attempt to break through the location privacy (in terms of cell location) of a user without his awareness. Cloaking techniques in other standard location privacy frameworks are not resistant against this attack as well. It can be best illustrated with Figure 3.7(a) (b).

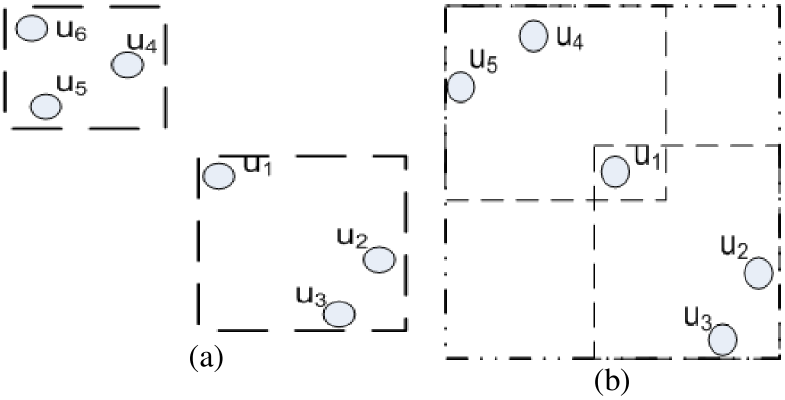


Figure 3.7. Attack scenario 2

As evident from the figures, there exists two 3-CRs with two sets of users $\{u_1, u_2, u_3\}$ and $\{u_4, u_5, u_6\}$. When the user u_6 leaves, the CRs are updated and new CR consisting of the users $\{u_4, u_5\}$ also includes u_1. So, the user u_1 is in the intersection of the two ASs consisting of two sets of users $\{u_1, u_2, u_3\}$ and $\{u_1, u_4, u_5\}$. Apparently, two subsequent requests from these two groups of users will reveal the actual location of the user u_1 to the attacker.

3.6 Remedy

Attack scenario 1 can be handled if the users in the same set use same CR over the time i.e. a cloaking algorithm that conforms to the reciprocity condition [Ghinita07, Kalnis07, Talukder10]. The users in a CR always use the same CR until a rearrangement of the CR takes place due to arrival or departure of some user(s). Hilbert Cloaking [Ghinita07, Kalnis07] is the one that meets such requirements but it performs poorly when it comes to CR area optimization. Similarly to avoid Attack scenario 2 the users can't use overlapping ASs. The ASs should be defined for disjoint groups of user sets and therefore conform to the reciprocity requirements. In following section we come up with an approach which meets reciprocity condition. Prior to that, a detailed definition of reciprocity condition is worthwhile.

Definition 1. The Reciprocity Condition

It is a condition for a cloaking technique where it is necessary that the LA generates the same CR for every user in an anonymity set for the same anonymity level. Consider that the LA generates a k-CR for the user u_1 which also includes k-1 other users $\{u_2, \dots u_k\}$. The reciprocity condition necessitates that if any other user apart from u_1, (i.e., a user having an index $1 < i \leq k$) issues any request, the CR of the issuer must be same as that of u_1. A cloaking algorithm is said to satisfy the reciprocity condition if the CR generated by the algorithm ensures that the set of users inside that CR will generate the same CR. Although not considered as a necessary condition for any of the cloaking techniques in the literature [Gruteser03, Gedik05, Mokbel06, Kalnis07], the attacks demonstrated in this section emphasize the importance of this consideration. The reciprocity condition has been addressed in Ghinita et al.'s work [Kalnis07] by proposing a cloaking scheme (besides Nearest Neighborhood Cloaking - NNC) namely Hilbert Cloaking which takes advantage of Hilbert space filling curve. Still, the technique suffers from the attack and query quality problem, as discussed in the evaluation section. Finally, we present our eventual solution designed to withstand the attack scenarios by constructing cloaked regions that satisfy the reciprocity condition with little overhead.

3.7 Enhanced SafeGrid

To ensure that our CR construction algorithm meets reciprocity condition we have made some changes to our basic approach. We partition entire user set into some groups which we call Anonymity Group ($GAnon$) where all users with identical anonymity requirement are placed in same $GAnon$. Anonymity Sets (AS) are formed by taking at least k no of users from each $GAnon$. Same AS and corresponding CR is used for each member of this set. The CR construction algorithm consists of the following steps.

Step 1. *Partition $GridTable$ based on distinct values of k to construct Anonymity Groups $GAnon_1$, $GAnon_2$, …, $GAnon_N$ where $GAnon_1$consists of all users with maximum k requirement and $GAnon_N$ consists of all users with minimum k requirement. η_i denotes the common k required by all users of $GAnon_i$.*

Step 2. *For each Anonymity Group Compute the distinct cells in that group.*

Step 3. *Select successive unassigned cells from $GAnon_1$until cumulative no of users in the selected cells reaches η_1. If cumulative no of users in all unassigned cells of $GAnon_i$ can't meet η_i, continue selecting unassigned cells from higher indexed $GAnon$ (i.e. $GAnon_{i+1}$).*

Step 4. *Mark all these selected cells in all Anonymity Groups as "assigned".*

Step 5. *Construct a CR with these cells.*

Step 6. *Update $GridTable$ by assigning this CR to each user of the cells in the CR.*

Step 7. *Repeat Steps 3-6 for $GAnon_1$, $GAnon_2$, ..., $GAnon_N$ until every cell is assigned into some CR.*

Details of the tasks accomplished in each of the above mentioned steps are shown in the following procedure.

SafeGrid-CR-CONSTRUCT (*GT*)

```
Sort $GT$ in descending order of $k$
$j \leftarrow 1$, $Anon \leftarrow GA[j].k$
for $i \leftarrow 1$ to $N$
    do if $GT[i].k = Anon$
        then $GA[j] \leftarrow GT[i]$
        else $j$++
            $GA[j].k \leftarrow GT[i].k$
            $Anon \leftarrow GA[j].k$
for $j \leftarrow 1$ to $n$
    do for $k \leftarrow 1$ to $length[GA[j]]$
        do if $GA[j][k].c$ not in $GA[j].cell$
            then $GA[j].cell \leftarrow GA[j].[k].c$
while $Cell$ has more elements
    do $Cell[c].locked \leftarrow False$
$CR \leftarrow Nil$, $i \leftarrow 1$
for $j \leftarrow 1$ to $n$
    do $Anon \leftarrow GA[j].k$
        while $CR[i].count < Anon$
            do while $GA[j]$ has more elements
                do $x \leftarrow GA[j][q].c$
                    if $x.locked = False$
                        then $CR[i] \leftarrow x$
                            $CR[i].count$ += $x.count$
                            $x.locked \leftarrow True$
return $CR$
```

The procedure takes the Grid Table, *GT* as input. Each entry of *GT* stores the current cell and anonymity requirement of each user as the values of *GT*[*i*].*c* and *GT*[*i*].*k* respectively. *GA*[*p*] is the *p*th anonymity group consisting of users having same anonymity requirement denoted by *GA*[*p*].*k*. The two **for** loops of lines 3-12 partition the *GT* into a finite number of *GA*'s along with computing the user count per cell.

Then in lines 13-14 each cell is initialized as unassigned/unlocked. The variable *Cell* holds a complete list of all those cells along with their availability information. The actual task of cloaked region construction occurs inside the **for** loop of lines 16-24. It picks unlocked cells from subsequent *GA*'s and puts them into a *CR* until the maximum requirement as denoted by *CR*[*i*].*count* is reached. Finally, the set of cloaked regions, *CR* is returned.

3.7.1 Illustrative Example

An illustrative example demonstrates, in detail, actually how an *AS* and the corresponding *CR* are constructed by the SafeGrid-CR-CONSTRUCT procedure. Figure 3.8 and 3.9 show a sample user distribution and the resultant initial state of *GT*.

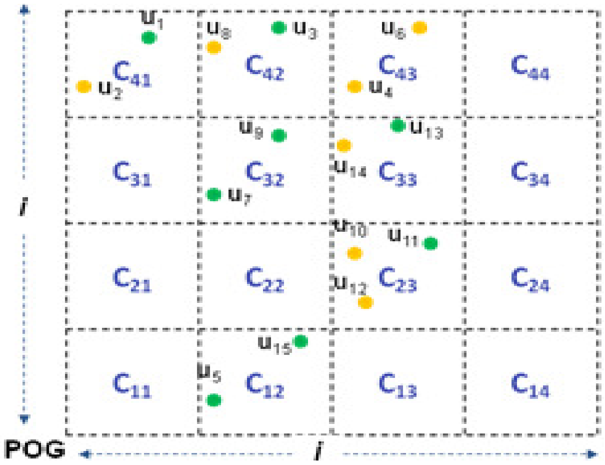


Figure 3.8. User Distribution in Location Grid

k = 3		k = 4	
User	Cell	User	Cell
u_1	C_{41}	u_2	C_{41}
u_3	C_{42}	u_4	C_{43}
u_5	C_{12}	u_6	C_{43}
u_7	C_{43}	u_8	C_{41}
u_9	C_{32}	u_{10}	C_{33}
u_{11}	C_{33}	u_{12}	C_{33}
u_{13}	C_{33}	u_{14}	C_{33}
u_{15}	C_{12}		

Figure 3.9. Initial *GridTable*

Figure 3.10 shows the changes of GT after each major step of the algorithm. Step 1 and Step 2 illustrates the process of partitioning GT into different GAnon's. Then the iterations of the CR construcion step is divided into sequential substeps. They exhibit the allocation of unlocked cells into subsequent CR's.

Step 1:

GAnon$_1$	<u$_2$, C$_{41}$>, <u$_4$, C$_{43}$>, <u$_6$, C$_{43}$>,<u$_8$,C$_{42}$>,
(k = 4)	<u$_{10}$, C$_{23}$>, <u$_{12}$, C$_{23}$>, <u$_{14}$, C$_{32}$>
GAnon$_2$	<u$_1$, C$_{41}$>, <u$_3$, C$_{42}$>, <u$_5$, C$_{12}$>,<u$_7$,C$_{33}$>,
(k = 3)	<u$_9$, C$_{32}$>, <u$_{11}$, C$_{23}$>, <u$_{13}$,C$_{33}$>,<u$_{15}$,C$_{12}$>

Step 2:

GAnon$_1$	C$_{23}$	C$_{33}$	C$_{41}$	C$_{42}$	C$_{43}$
(k = 4)	(3)	(2)	(2)	(2)	(2)
GAnon$_2$	C$_{23}$	C$_{32}$	C$_{12}$	C$_{41}$	C$_{42}$
(k = 3)	(3)	(2)	(2)	(2)	(2)

Step 3.1:

CR$_1$: [C$_{23}$, C$_{33}$]

GAnon$_1$	C$_{23}$	C$_{33}$	C$_{41}$	C$_{42}$	C$_{43}$
(k = 4)	(3)	(2)	(2)	(2)	(2)
GAnon$_2$	C$_{23}$	C$_{32}$	C$_{12}$	C$_{41}$	C$_{42}$
(k = 3)	(3)	(2)	(2)	(2)	(2)

Step 3.2:

CR$_1$: [C$_{23}$, C$_{33}$]; CR$_2$: [C$_{41}$, C$_{42}$]

GAnon$_1$	C$_{23}$	C$_{33}$	C$_{41}$	C$_{42}$	C$_{43}$
(k = 4)	(3)	(2)	(2)	(2)	(2)
GAnon$_2$	C$_{23}$	C$_{32}$	C$_{12}$	C$_{41}$	C$_{42}$
(k = 3)	(3)	(2)	(2)	(2)	(2)

Step 3.3:

CR$_1$: [C$_{23}$, C$_{33}$]; CR$_2$: [C$_{41}$, C$_{42}$]; CR$_3$: [C$_{43}$, C$_{32}$, C$_{12}$]

GAnon$_1$	C$_{23}$	C$_{33}$	C$_{41}$	C$_{42}$	C$_{43}$
(k = 4)	(3)	(2)	(2)	(2)	(2)
GAnon$_2$	C$_{23}$	C$_{32}$	C$_{12}$	C$_{41}$	C$_{42}$
(k = 3)	(3)	(2)	(2)	(2)	(2)

Figure 3.10. Steps of CR Construction

Finally, the constructed anonymity sets and cloaked regions are shown in Figure 3.11. From Figure 3.11(b) it is evident that each user experiences greater level of anonymity than their requested ones.

Cloaked Region	Anonymization Set	Achieved Anon.
CR_1: [C_{23}, C_{33}]	AS_1: [u_{10}, u_{11}, u_{12}, u_{13}, u_{14}]	5
CR_2: [C_{41}, C_{42}]	AS_2: [u_1, u_2, u_3, u_8]	4
CR_3: [C_{43}, C_{32}, C_{12}]	AS_3: [u_4, u_5, u_6, u_7, u_9, u_{15}]	6

(a) Constructed CRs and corresponding disjoint *AS*s

User	u_1	u_2	u_3	u_4	u_5	u_6	u_7	…	u_{15}
***k* (reqd.)**	3	4	3	4	2	4	3	…	3
Cell	C_{41}	C_{41}	C_{42}	C_{43}	C_{12}	C_{43}	C_{43}	…	C_{12}
CR	CR_2	CR_2	CR_2	CR_3	CR_3	CR_3	CR_3	…	CR_3
***k* (achd.)**	4	4	4	6	6	6	6	…	6

(b) Updated *GridTable* stored by the LA

Figure 3.11 Final Outcome of SafeGrid-CR-CONSTRUCT

However, this illustrative example does not hold sufficient proof to our claim of satisfying the reciprocity condition. A theoretical proof on that follows next.

3.7.2 Resolving Trade-off between privacy and QOS

After constructing the CR Location Anonymizer modifies the actual query by replacing the $Cell\_ID$ of actual requester with the set of cells belonging to the CR. At this point, LSP receives the $Cell\_List$ along with the proximity query. It then returns the candidate list consisting of possible query results to the LA. Query results are eventually returned to the actual query issuer in a special tabular format which includes the $Cell\_ID$ of each POI. Upon receiving the list of possible candidates user finds the closest one. A noteworthy feature of SafeGrid is that computation overhead in client side remains almost same with increasing value of k. In order to facilitate this we make use of efficient data structure and smart techniques of message passing which are shown below.

$Pseudo\_ID$	$Query$	$CR\ (Cell\_List)$

(a) Request Message Format

$Cell\_ID$	$Query\ Candidates\ for\ Cell\_ID$

(b) Response Message Format

C_{11}	q_1, q_2	$\dots$	C_{ij}	q_i, q_{i+1}	$\dots$	C_{mn}	q_m, q_{m+1}

(c) Response Message Data Structure

Figure 3.12 Message Formats and Data Structure

Query results are returned to the user according to the above mentioned format. Depending on value of k size of this transferred data may grow or shrink. But the important point to note that user is concerned only about the query results lying in his own cell, say $\boldsymbol{C_{ij}}$, and ignores all other results. Thus a user's actual query processing time depends on the size of his own cell size disregarding the size of CR generated by the LA. So, processing time at client side is unaffected by his anonymity requirement. This is an indication that SafeGrid greatly reduces the trade-off between privacy and user side QOS.

3.7.3 Proof of Attack Resistance

In order to prove the capability of our proposed approach to resist the attacks mentioned in this paper, we show that the cloaked region construction algorithm meets reciprocity condition.

Theorem: *SafeGrid-CR-CONSTRUCT meets the reciprocity condition.*

Proof: u_m submits a query and the following cloaked region is constructed.

$CR(u_m) = \bigcup_{i=1}^{k} c_i$ where $k \leq n$, n is total no. of cells

$AS(u_m) = \bigcup_{i=1}^{k} u_{m_i}$ where k is requested anonymity.

Let, $u_n \in AS(u_m)$ then, $u_n.c \in CR(u_m)$

We prove that, $CR(u_n) \subseteq CR(u_m)$

Suppose, $c_x \in CR(u_n)$ ... (1)

Now, since $u_n \in AS(u_m)$

$\Longrightarrow u_n.k = u_m.k$ … (2) [Acc. to CR constr. algorithm]

Then, $(u_n \in GA_p) \Longrightarrow (u_m \in GA_p)$ [Defn. of GA]

where the pth anonymity group, $GA_p = \bigcup_{y=1}^{u_n.k} u_y$

$CR(u_n) = \{\cup\, c_i \mid \forall y\, \exists i\; c_i = u_y.c\}$

Now, $\forall c_x \in CR(u_n)\; \forall u_i \in c_x\; u_i.k = u_n.k$

Let, $c_x = u_q.c$ and $u_q.k = u_n.k = \lambda$

According to Eqn. (2) $u_q.k = u_m.k$

$\Longrightarrow AS(u_q) = AS(u_m)$

$\Longrightarrow u_m \in AS(u_q)$

$\Longrightarrow c_x \in CR(u_q)$

$\Longrightarrow c_x \in CR(u_m)$ … (3)

$[\forall u_j \in AS\big(u_q\big)\; c_x CR(u_q) \Longrightarrow c_x \in CR(u_j)]$

Hence, from eqn. (1) and eqn. (3) we obtain,

$\forall x\; c_x \in CR(u_n) \Longrightarrow c_x \in CR(u_m)$

$\Longrightarrow CR(u_n) \subseteq CR(u_m)$ … (4)

It can be shown that (4) is true for any other user lying inside the cloaked region of u_m. So, any other user with same anonymity requirement as u_m yields cloaked region which does not contain any cell that is not inside the cloaked region of u_m. Thus, *SafeGrid-CR-CONSTRUCT* meets the reciprocity condition. ◊

Chapter 4: Identity Inference Protection

Our proposed location privacy framework, SafeGrid enhances the attack resistance capability. It ensures privacy of location information against a wide variety of attacks including the novel ones introduced in this thesis. The major feature of satisfying the reciprocity condition makes it superior to the existing CR-based approaches. However, the ultimate goal of preserving privacy lies in preventing unintentional inference of the identity of actual query issuer. In this regard, our basic system, along with all the other ones found in current literature, fail to provide a full-proof solution, atleast in the sense that under some situations the disclosure of the identity of the user may be highly likely. In this chapter, we focus on such more advanced dimension of privacy protection which deals with identity revelation threats stemming from location based inferences. First, we show how the current measure of *k*-anonymity is not an adequate one considering its vulnerability to several new attacks. Afterwards, we present the definition of a new measure, called *s*-proximity. Finally, we propose slight modifications to the earlier proposed framework mostly in terms of augmenting more capabilities and functionalities on the part of the trusted third party. In addition, several innovative algorithms are formulated to assimilate the proposed privacy measure into the cloaking algorithm being used.

4.1 Identity Inference Attacks

In this section, we exhibit that the conventional measure, *k*-anonymity does not provide sufficient protection against privacy violation. Two attacks called, the *heterogeneity attack* and the *conformity attack*, are presented and it is demonstrated how they can be used to compromise a *k*-anonymous location based query. The heterogeneity attack reveals that *k*-anonymity can create groups that fail to provide overall anonymity due to lack of sufficient match among the members with respect to some sensitive user attribute. Likewise, approaches satisfying only *k*-anonymity disregard consequences of revealing important context [Talukder08] information though the service request and pave the way for conformity attack. Besides illustrating the attacks with real world examples, the section provides their formal definitions which clarify how they relate to the contexts [Talukder08] of the query and static information [Machanavajjhala06] of the users in the anonymity set (AS) [Machanavajjhala06].

4.1.1 Example Scenarios

Let's take a look at some real world examples of location based services to better understand the threat of privacy violation. The examples demonstrate that existing solutions which depend on satisfying *k*-anonymity are still vulnerable to privacy violation attacks.

We assume Alice is currently subscribed to a location based system which uses a trusted LA to make her location *k*-anonymous and forward her query to the LSP. Alice is guaranteed that her exact location is never disclosed to the LSP by assuring that she remains *k*-anonymous to the unknown third-party where *k* is chosen by herself according to her required privacy level. However, we present couple of random scenarios where her ultimate privacy is shown to be endangered although her location is *k*-anonymous to the LSP.

Scenario 1: Alice, owing to some chronic disease, goes through regular medical checkup. She moves to a new place and looks for the nearest medical center. Due to the specific nature of her illness she is considering only healthcare centers that treat feminine diseases. This makes it logical for her to query for the nearest female hospital from her current location. She submits her query regarding locating the nearest female hospital to the LSP. The LBS system that handles her request provides her with the requested service along with preserving her location privacy by means of ensuring that her location information is made *k*-anonymous before being submitted to any un-trusted party (in this case, LSP).

Scenario 2: Here Alice uses the LBS system for her academic purpose. As she starts her new semester in the graduate school, after the first week of classes she is given a list of books some of which she needs to purchase urgently. She searches for the nearest bookstores from her residence and is served by the LBS system. The LBS system takes her location and returns her the list of book stores located nearest to her place.

Scenario 3: Alice makes frequent travels to new places where she faces the problem of finding nearest car parks. She is reluctant to disclose all the places she visits due to privacy concern. Hence she gets her privacy-aware LBS system involved into the task of finding nearest car parks. She feels comfortable because she gets the service without disclosing either her identity or location data.

In the above scenarios Alice has to provide her location to get the services but due to threat of identity disclosure she is reluctant to provide her exact location, rather the LA creates a CR and forwards her query to the LSP. If Alice has, for example, *k*=4 anonymity requirement, the LA makes sure that instead of her location a CR that contains at least 3 other users is sent to the LSP. Thus the LBS system tries to protect her location privacy and finally preserve any un-solicited identity disclosure. Still her identity can be disclosed to the LSP as discussed below.

4.1.2 Attacks on *k*-Anonymity

We assume that the LSP or any other adversary has access to the following information [Kalnis07].

1) Accesses requested anonymity level (i.e. value of *k*) for each query.

2) Recognizes the users belonging to the AS corresponding to the CR of a query.

3) Collects static profile data of all the users that have made query at some time.

4) Identifies any special context revealed by the query and maps that context to relevant user attribute.

Finally it is supposed that a group of malicious LSPs collaborate with each other by sharing their knowledge about the users in an effort to re-identify any individual query requester. Based on the above assumptions we summarize the knowledge of the LSPs involved in the LBS system in the following table.

TABLE 4.1 IDENTITY INFERENCE ATTEMPT FROM BACKGROUND KNOWLEDGE

Criteria	**Scenario 1**	**Scenario 2**	**Scenario 3**
Query	Nearest Female Hospital	Nearest Book Store	Nearest Car Park
LSP	LSP_1	LSP_2	LSP_3
|AS| (value of k)	4	4	4
Members of AS (LSP has identified)	{Alice, Bob, William, Ada}	{Alice, Carl, Jacob, Michael}	{Alice, Bob, Daniel, Joshua}
Context of the service	Healthcare	Academic	Transportation
Relevant user attribute	Gender	Occupation	Driving License
Findings	Only Alice is female in the group	Only Alice is student in the group	Only {Alice, Joshua} have Driving License
Identity Inference (probable query requester)	Alice	Alice	{Alice, Joshua}

In Scenario 1 the attacker (in this case LSP_1) successfully identifies Alice to be the query issuer. The fact that she was grouped with 3 male users who seem unlikely to request for a female hospital enabled the re-identification. Her query was too specific as well. In Scenario 2 the attacker succeeds again. Here Alice is grouped with non academic users which made the attacker guess Alice to be the query requester. In Scenario 3 the attacker was able to narrow down the list of possible issuers. From the context of the service being requested the attacker knows that any user not having driving license is unlikely to request for a car park. There might be numerous occasions such as discussed above where *k*-anonymity fails to guard against identity disclosure. We have classified all these into two categories which are discussed below.

4.1.3 Classification of Attacks

1) Heterogeneity Attack: The members in the anonymity set are too much diversified with respect to some static attributes. In worst case, the query requester might possess some exclusive identifiable property, then she no longer remains indistinguishable. For example, an AS which groups a female user with all male users is vulnerable to such attack.

Observation: The anonymity set should ensure the inclusion of a minimum number of users with similar profile as the actual query requester with respect to some static attributes.

2) Conformity Attack: The service being requested in the query is too specialized as it relates to some particular contexts. A user has to conform to some specific conditions to be a potential candidate for such service. If most of the users in the AS fail to possess those properties, chances of the actual requester being re-identified increase. For example, a query for a nearest female hospital is too specific. Instead the request for a nearest hospital involves fewer contexts [Talukder08, Machanavajjhala06].

Observation: The query is made more generalized by changing granularity level [Machanavajjhala06] of the intended service.

Both of these two types of attacks have some common properties as they relate to the context of the query and the static attribute of the requester. To better understand the notion of this novel attack we define it formally in the following section.

4.2 Attack Model

Now we move into the formalization of the attacks. However, prior to that a walk through the notations and definitions will be worthwhile.

4.2.1 Definitions

User Set (U): The LA maintains a list of users who have subscribed to it. This list is denoted by $U = \{u_1 , u_2, \ldots , u_N \}$ where u_i represents ith user.

Context (C): Any sensed information used to describe some physical phenomena is defined as context [Talukder08]. In this paper we mean by context any deterministic condition or situation that characterizes a service. The contexts can take on different levels of granular values. A finite domain of information for all the contexts in the application is assumed in our proposed model. Thus, the individual sets of contexts will have a finite number of possible values. Using higher level granular values the service becomes more generalized.

Service (S): We consider, the set $S = \{s_1, s_2, \dots, s_m\}$ consists of location based services only. A user has to provide some sort of location data to avail such services. Although location is the major emphasized context for these services, other contexts are also associated with each service. Generally each service is provided by an LSP whereas it is common for an LSP to deliver a group of services.

Static Information (SI): The static information could be the user's static attributes or credentials used to authenticate his identity. We refer to the set of static information as $SI = \{si_1, si_2, \dots, si_k\}$. We use the terms *static information*, *static user attribute* and *static user profile* interchangeably throughout the paper. In some places we have used *SI* = $\{a_1, a_2, a_3, \dots, a_k$ *}* where a_i stands for *i*th *static user attribute.* Values of these attributes specify individual query requesters. A subset of *SI* forms quasi identifier [Machanavajjhala06, Cuellar02]. The *static information* is not directly provided by the requester of a location based query rather an attacker collects it from background data sources.

Anonymity Set (AS): The list of probable issuers of a query request is called an anonymity set. The request can be issued by any candidate in the anonymity set. The re-identification process becomes more difficult as the cardinality of the set increases

4.2.2 Formal Definition of the Attack

The attacker is supposed to possess following knowledge [Mokbel06, Talukder08, Ghinita07].

1. Query Request, r: (s, c, si) where

$s \in S$: service being requested; $c \subseteq C$: contexts of the service; $si \subseteq SI$: relevant static user attributes corresponding to the contexts of the service

2. Cloaked Region, $CR = \{(x, y) | x, y \in R \wedge (CR.w_1 \leq x \leq CR.w_2) \wedge (CR.h_1 \leq y \leq CR.h_2)\}$

3. Requested Anonymity level, k

4. The knowledge of CR and k together suffice to know about the anonymity set.

Anonymity Set, $AS = \{u_1, u_2, \ldots, u_k\}$, where $u_i \in U \; \forall i$

Now, we suppose the following:

$s_j.v(c_i)$: Value of the context c_i of service s_j

$u_r.v(si[c_i])$: Value of the static user attribute corresponding to context c_i of user u_r

Based on the above assumptions we formally define the attack model below.

Suppose, u_r submits query $r = (s, c, si)$ to the LSP with cloaked region formed by $AS = \{u_1, u_2, \ldots, u_k\}$ to meet her k-anonymity requirement.

We assume that, u_r is distinctive from every other user in AS with respect to any static user attribute corresponding to some context of the service being requested.i.e., $\exists c_i \in c | u_r.v(si[c_i]) \neq u_j.v(si[c_i]), \forall j \neq r$ ... (1)

Now, the LSP being a malicious attacker finds out some context c_a relevant to query r and si_a be a static user attribute corresponding to c_a. Then it looks up in background data sources to collect data of si_a attribute for all $u_i \in AS$. Since u_r is the actual query requester, $u_r.v(si[c_a]) = s.v(c_a)$ ... (2) holds true.

According to Assumption (1):

$\exists j | u_r.v(si[c_a]) \neq u_j.v(si[c_a]), \; j \neq r$ ... (3).

The attacker can exclude from the AS the users that satisfy eqn. (3). As the size of AS shrinks, the probability of re-identifying u_r increases. Collaboration among a group of malicious LSPs may yield a context c_x for which eqn. (2) may hold for every $u_i \in AS$ except u_r. Then u_r could be immediately re-identified as the query issuer.◊

As we have already defined the identity inference attack in a formal way, we proceed to introduce our approach to handle it. At the core of our solution there exists a new privacy parameter called *s-proximity* which we introduce next.

4.3 s-Proximity: A New Privacy Parameter

A close look at the attack scenarios reveals that if actual query requester is fully distinguishable from other users in the AS, with respect to some static attribute relevant to the context of the query, her identity may be disclosed immediately whatever her achieved location anonymity may be. Therefore, it is highly desirable that at least a minimum number of users in the AS have similar profile as the actual query issuer. This requirement adds a novel parameter, called s-proximity, to the users' privacy profile [Li07]. Before defining s-proximity, we need to introduce couple of relevant definitions.

Dissimilarity Measure: This metric measures the amount of divergence between two users with respect to a certain static user attribute. We use the notation $d_{si_p}(u_m, u_n)$ to denote the dissimilarity measure between u_m and u_n based on si_p where si_p is a static attribute corresponding to the context c_p of service s_j. $d_{si_p}(u_m, u_n) < \delta \Rightarrow u_m \sim u_n$ [u_m is "similar to" u_n], δ is a user defined threshold value.

Equivalence Class (E): The set of users that are similar to $u_i$with respect to pth static attribute si_p is called equivalence class of u_i and denoted by $E_{si_p}(u_i)$.

$$E_{si_p}(u_i) = \{u_j \in U | (u_i \sim u_j) \wedge (si = si_p)\}$$

$$= \{u_j \in U | d_{si_p}(u_i, u_j) < \delta\}$$

Divergence Class (D): The set of users that are not similar to $u_i$with respect to *p*th static attribute si_p is called divergence class of u_i and denoted by $D_{si_p}(u_i)$.

$$D_{si_p}(u_i) = \{u_j \in U | (u_i \nsim u_j) \wedge (si = si_p)\}$$

$$= \{u_j \in U | d_{si_p}(u_i, u_j) \geq \delta\}$$

s-Proximity: By *s-proximity* we mean that the AS will contain at least (*s-1)* other users similar to the actual query requester u_r, i.e., $|AS \cap E_{si_p}(u_r)| \geq s$.

Selecting higher value of *s* guarantees strong privacy but at the cost of degraded quality of service. So, users themselves are responsible for choosing value of *s* according to their preference.

Enhanced Privacy Profile ($\boldsymbol{k, s, A_{min}}$): With the introduction of *s-proximity* our model assumes that a user's privacy profile consists of k, s and A_{min} which stand for anonymity requirement, proximity requirement and minimum CR area constraint.

4.4 Context-Aware Location Anonymizer

We start our discussion with the generalized view of a location privacy framework having a trusted location anonymizer (LA). In such systems subscribed users send their location based query to the LA which replaces the exact location with a cloaked region and forwards the query to the LSP. In reply LSP returns the list of query results which is usually termed as list of point of interests (POI) to the LA and eventually the POI list is forwarded to the query requester. The system is depicted in the figure below.

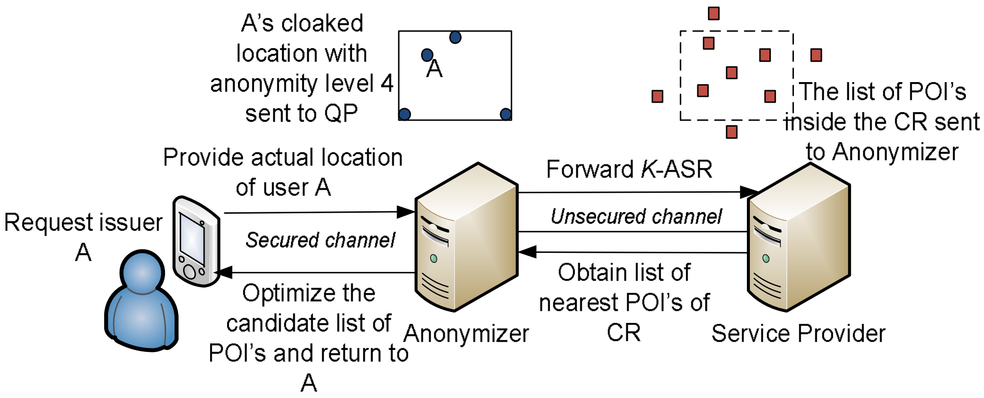


Figure 4.1. The Location query processing through LA

Such a system tries to preserve user's location privacy by implementing the measure of *k*-anonymity. As long as a system ensures that user's location is *k*-anonymous to the LSP, it apparently succeeds in preserving location privacy. In this paper we have shown, this typical notion of safeguarding location privacy by means of *k*-anonymity is not adequate rather it may endanger a user's ultimate privacy by revealing her sensitive private data. To provide a robust privacy solution we need to ensure both *k*-anonymity and *s*-proximity. We use this existing privacy framework and incorporate advanced functionalities into the LA to provide such a solution.

4.4.1 Overview of the System

We propose a location privacy system with a trusted LA which creates anonymization group considering context of the query. At the core of our approach lies an enhanced location anonymizer attributed with multiple capabilities and we call it a *Context Aware Location Anonymizer (c-LA)*. As the basic functionality is same we use the terms *LA* and *c-LA* interchangeably hereinafter. The main focus of the solution is on minimizing the probability of re-identifying the actual query requester along with anonymizing his location information. A location based query usually contains other sensitive information alongside spatial data. Hence, it is not enough to hide only the location data rather we propose modifying the query as a whole to minimize any identifying information it carries. We term this process *Query Generalization* which is the first step of our solution. The task is accomplished at the LA by an additional module called *Query Analyzer* which identifies any sensitive context in the query and looks for possible generalization. Only generalizing the query does not solve all the problems. As we discussed earlier the way users are grouped together to form anonymization set impacts the possibility of re-identifying actual query issuer. The task of satisfying the *s-proximity* condition is performed by the module called *Partitioning Agent* responsible for splitting the entire user set into *Equivalence Class* and *Divergence Class.* Finally, the *CR Construction Unit* generates the cloaked region based on anonymization set created from users in the *Equivalence Class.*

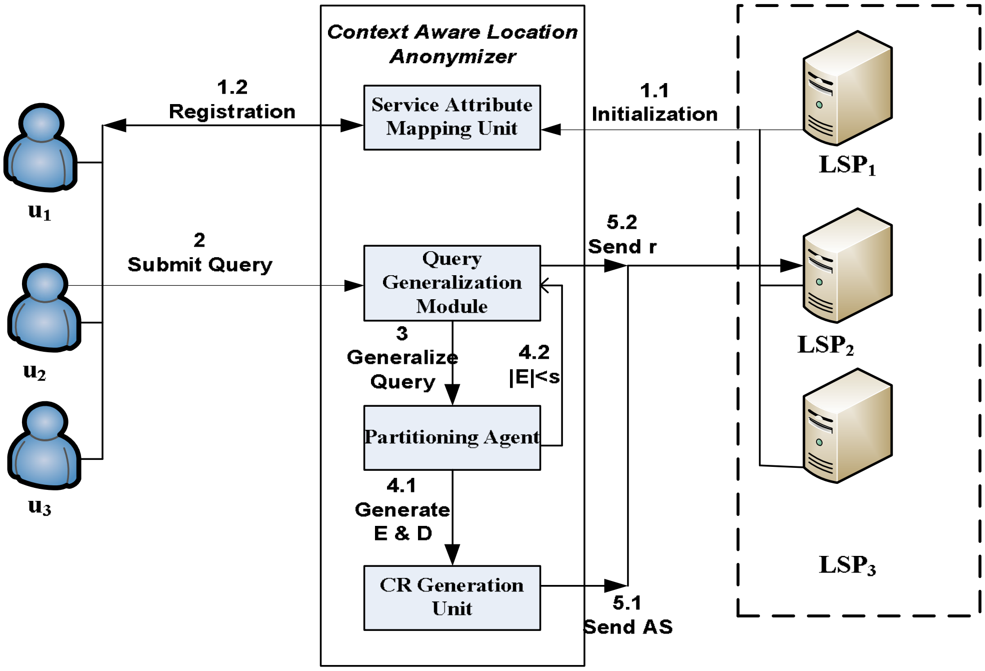


Figure 4.2. Architecture of Context Aware Location Anonymizer ***(c-LA)***

4.4.2 Details of Our Solution

The *Context Aware Location Anonymizer,* as depicted in Figure 4.2, consists of multiple units, each unit performing dedicated tasks. The overall process of anonymization is accomplished in several steps which are discussed below in detail.

Step 1. Initialization: Location Anonymizer maintains a list of services for which it has registered with corresponding LSPs. For each service, LA has knowledge of relevant contexts using which it generates a set of static information interrelated with the service. LA stores all these information in the *Service Attribute Mapping (SAM)* table which has the form: $< Service, Context\_List, Static\_Information\_List >$

The LA informs its subscribed users about its *SAM* table. A user chooses the services of his interest and finds the set of required static information. The user then registers with the LA providing that static information.

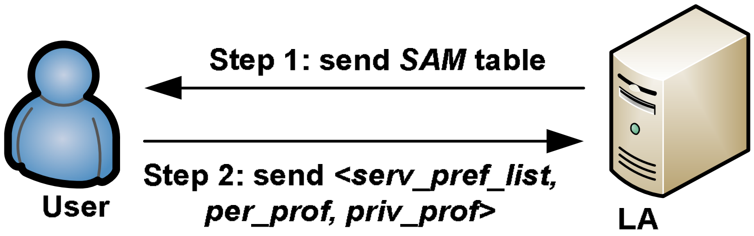


Figure 4.3. Initialization Phase

At the time of registration the user also notifies the LA about his privacy profile. The user does that in Step 2 in the above figure by passing the values of $< k, s, Amin >$ denoted by $priv\_prof$. In the same message the user includes his personal information in per_prof and his preferred services list in $serv\_pref\_list$. At the end of this phase the LA gets necessary information from the subscribed users to fill in the *User Profile* table. Structure of this table is: $< User, serv\_pref\_list, per\_prof, priv\_prof >$

Step 2. Query Generalization: In this step the LA tries to generalize the query by modifying the content of the request. In effect, it modifies the granularity level of the requested service. To facilitate that all the services are arranged in a hierarchical tree structure where a node represents the generalized service for all the services in the sub-tree rooted at that node. The contexts related to the services are used to determine the correlations among them and to construct the tree. This step may be termed as the ***Service Generalization*** step as well. The main task of service generalization is accomplished by using a generalization function. In order to better understand how the generalization function works we intend to define the query request in a formal way as follows.

A query request consists of the identity of the user, the context and the static information and it has a complex domain: $R \in \{U \times \Pi_i\, C_i \times \Pi_\mathrm{j}\, S_j\}$. The context and static information contained in this request are modified by the generalization function, G_s to yield a generalized request. The generalized request domain can be represented as: $R' \in \{U' \times \Pi_\mathrm{i} C_i' \times \Pi_\mathrm{j}\, S_j'\}$ which contains the context and static information augmented to higher granularity levels. We have identified the following properties [Talukder08] which the generalization function $G_s: R \rightarrow R'$ holds.

1. Many to one mapping: Two or more requests can be transformed into same augmented request that is forwarded to the service provider. $\exists r_1, r_2 \in R, G_s(r_1) = G_s(r_2) = r', r_1 \neq r_2$

2. Idempotent with generalized request: If the generalization function is applied to a generalized request, no more generalization will be possible, provided the generalization criteria along with anonymity and proximity levels remain the same. $\forall r \in R, G_s^n\big(G_s(r)\big) = G_s(r), n = 1, 2, \dots$

3. Invertible: The generalization function also has an inverse function. $\forall r \in R, \exists r_1 : G_s^{-1}(r) = r_1, G_s^{-1} : R' \rightarrow R^n, n \geq 1$. When $n = 1$, the issuer of the request is re-identified hence, the inverse generalization function can be used in simulating inference attacks.

4. Asymmetric: The generalization function and its inverse are asymmetric in nature. $\forall r \in R : G_s^{-1}(G_s(r)) \neq r,$ where $G_s^{-1} : R' \rightarrow R^n, n \geq 1$

5. Non injective: The generalization function is non injective in nature. If two or more requests are generalized using the same augmented request it doesn't imply that the requests are the same. $\forall r_1, r_2 \in R : G_s(r_1) = G_s(r_2) \nRightarrow r_1 = r_2$

6. Non Equivalence of function: The generalization functions having different privacy preferences may provide the same generalization for two or more requests. $\forall r \in R : G_s^1(r) = G_s^2(r) \nRightarrow G_s^1 \cong G_s^2$. Although some of the requests achieve same generalizations, G may not be equivalent due to the fact that all of the context or static values are not assigned, or the granularity level is coarse enough so that the generalization was not applied even.

Step 3. Proximity Group Formation: The user submits his location based query along with her privacy preference parameters (k, s, A_{min}) to the LA. In our proposed model user has option to send a set of additional parameters which set the priorities of the static information variables involved with the service being requested. Based on these inputs the LA partitions the entire user set into two disjoint subsets: *Equivalence Class* and *Divergence Class* according to the following algorithm.

Algorithm: Proximity Group Formation

```
Input: query $r \in R: < s, c, si >$, weight matrix $w$
Sort $si$ in descending order of $w$
for $(i = 1\ to\ k)$ do
    for $(j = 1\ to\ N)$ do
        if $(d(U[j].si[i], U[r].si[i]) \leq \delta)$ then
            Insert $U[j]$ into $E$
for $(j = 1\ to\ N)$ do
    if $(U[j] \notin E)$ then
        Insert $U[j]$ into $D$
return $E, D$
```

The algorithm takes the query, *r* and weight matrix, *w* as input. *w* contains user's preferred priority of the involved static attributes. Based on that priority other users are compared with the query requester. The users that are similar to query requester are inserted into equivalence class, *E* and others are inserted into divergence class, *D*.

Step 4. Cloaked Region Generation: In this step the LA constructs the cloaked region which is forwarded to the LSP. First, it chooses the AS in such a way so that it meets both *k*-anonymity and *s*-proximity. Then using the locations of the members of AS the CR is constructed which meets the A_{min} requirement. The CR generation process follows the algorithm presented below.

Algorithm: Selective Nearest Neighbor (SNN) Cloak

```
Input: $U, E, k, s, A_{min}, L$
Initialization: $AS = \{\}, i = 0$
Sort users in $U$ in ascending order of dissimilarity from $U[r]$
while $(|AS| < k)$ do
    if $(|AS| < s)$ then
        if $(U[i] \in E)$ then
        Insert $U[i]$ into $AS$
        $i$++
        else $i$++
    else Insert $U[i]$ into $AS$
call $GenerateCR(L, AS)$
return $CR$
```

The equivalence class of the query issuer, E, privacy profile (k, s, A_{min}) and location vector, L are supplied to the algorithm as input. It first tries to meet s-proximity by inserting s nearest users into AS taken from E. Then it inserts other *(k-s)* nearest users into the AS to meet k-anonymity. Finally the AS along with location vector, L is used to construct the CR.

4.4.3 Attack Prevention: Formal Proof

We conclude this section by showing that our CR generation algorithm SNN-Cloak is attack resistant.

Lemma: $c - LA$ reduces the probability of re-identification of actual query issuer.

Proof: Let, $u_r \in U$ submits query $r = (s, c, si)$ to the LA with her k-anonymity and s-proximity requirement. The LA constructs two different anonymity sets AS_1 and AS_2 applying SNN algorithm and NN algorithm [Kalnis07] respectively. AS_1 meets both k-anonymity and s-proximity requirement however AS_2 meets only k-anonymity.

Suppose, c_x be a context relevant to query r and si_x be a static user attribute corresponding to c_x. Since u_r is the actual query requester,

$u_r.v(si[c_x]) = s.v(c_x)$ ... (1)

Let, E_x be the equivalence class of u_r with respect to si_x. Then, $u_j.v(si[c_x]) = s.v(c_x), \forall u_j \in E_x \ldots (2)$.

We define, $E_{SNN} = E_x \cap AS_1$ and $E_{NN} = E_x \cap AS_2$. Hence, $u_j.v(si[c_x]) = s.v(c_x), \forall u_j \in AS_1 \ldots (3)$[SNN meets s-proximity] Conversely,

$\exists u_j \in AS_2 | \ u_j.v(si[c_x]) \neq s.v(c_x), \ldots (4)$ [NN does not meet s-proximity] Using (3) and (4) we get $|E_{SNN}| > |E_{NN}| \ldots (5)$.

In AS_1 the probability of re-identifying u_r based on c_x and $si_x = \frac{1}{|E_{SNN}|}$

Similarly, In AS_2 the probability of re-identifying u_r based on c_x and $si_x = \frac{1}{|E_{NN}|}$

From (5) $\Rightarrow \frac{1}{|E_{SNN}|} < \frac{1}{|E_{NN}|}$. So, the SNN algorithm applied by $c - LA$ reduces the probability of re-identification of actual query issuer.◊

Chapter 5: Related Work

Gruteser et al. [Gruteser03] were the first to introduce cloaking technique known as Interval Cloak (IC). They calculate CR based on a Quad-tree approach, where they recursively partition the space into four equal squared regions until the user fits in a quadrant where *k*-anonymity is satisfied. Another quad-tree variant using pyramid structure called New Casper by Mokbel et al. [Mokbel06] achieves superior worst-case complexity for calculating CR over IC [Gedik05]. Gedik et al. [Gedik05] introduced privacy personalization framework and developed a CR algorithm known as Clique Cloak (CC). It assigns Minimum bounding rectangle (MBR) for all the users and if the users' MBR intersects, they are eligible for forming a clique among themselves. *k*-anonymity is satisfied if the user belongs to a *k*-cliqued region. Hilbert Cloak (HC) introduced by Ghinita et al. [Ghinita07, Kalnis07] (HC) uses Hilbert's Space-Filling Curve to map user positions in 2-D space into 1-D values. These are subsequently partitioned into groups/buckets of *k* users. HC finds the group to which a user belongs, and returns the minimum bounding rectangle of the group as their CR. Nearest Neighbor Cloak (NNC) by Kalnis et al. [Kalnis07] tries to make CR small by taking the MBR of the nearest neighbors of a user. Bamba et al. [Bamba08] introduced PrivacyGrid (PG) based cloaking technique. They have a similar grid structure like ours. However, the grid is static and it does not consider users' preferences in choosing grid resolution. LA in PG knows the exact locations of users. The CR algorithm does not conform to reciprocity condition as it computes CR for each user individually. They construct CR by expanding in directions (North, South, East or West) by adding a row above the uppermost selected row (or below the lowermost selected row) or a column to the right of the rightmost column (or to the left of the leftmost column). In this process their CR may take cells where there is no user. Primary contribution of it is the introduction of *l*-diversity.

Matt Duckham et al. [Duckham05] first used an obfuscation technique without LA. The negotiation process proposed by them may run for several iterations causing prolonged service time and the more high QoS the user wants to avail the more private information (less degradation of his actual location) he needs to reveal. Techniques described in [Chow06, Kalnis07] eliminate anonymizer by considering mutual trust among peers. But forming trust relationship in an open dynamic environment could be an issue to begin with. The recent technique by Ghinita et al. [Ghinita08] uses a variant of Private Information Retrieval (PIR) theory known as Computational PIR (CPIR) for finding the approximate and exact Nearest neighbors of the Point of interests (POI). Due to the overwhelming computational time techniques using PIR theory seem to be infeasible for pervasive environment. CPIR used in [Ghinita08] also requires an additional overhead of a huge list of POIs to be sent to the resource constrained device and a malicious server may get into user’s private data though not in polynomial time. In [Ghinita08] LSP can modify the grid regions at its will as it has control over selecting granularity of grid design. Furthermore, [Ghinita08] imposes extra overhead in user side in two ways: a user has to perform numerous cryptographic computations and he has to maintain a secure communication channel. Following table summarizes the relevant research works.

TABLE 5.1 SUMMARY OF COMPARISON AMONG DIFFERENT CLOAKING APPROACHES

CR Approach	Generate CR for each user/query	Reveals exact location to LA	Requires secure communication with LA	Reciprocity Condition	Vulnerable to Attack 1	Vulnerable to Attack 2
IC [Gruteser03]	Yes	Yes	Yes	No	Yes	Yes
Casper[Mokbel06]	Yes	Yes	Yes	No	Yes	Yes
NNC[Kalnis07]	Yes	Yes	Yes	No	Yes	Yes
HC [Ghinita07]	No	Yes	Yes	Yes	Yes	No
PG [Bamba08]	Yes	Yes	Yes	No	Yes	Yes
SafeGrid	No	No	No	Yes	No	No

Recently Mascetti et al. investigated a more general case in which the adversary is able to recognize traces of LBS requests by the same anonymous user [Mascetti09]. They introduced the notion of *historical k-anonymity*. In [Xu09] Xu et al. presented a novel technique that allows a user's location information to be reported as accurate as possible while providing her sufficient location privacy protection. They also investigated the problem of preventing an adversary from locating nodes based on their location information they disclose in communications [Xu10]. They attempted to reduce location resolution to achieve a desired level of safety protection and presented a novel geographic routing protocol which can work with blurred location information. None of these works addressed the notion of reciprocity condition.

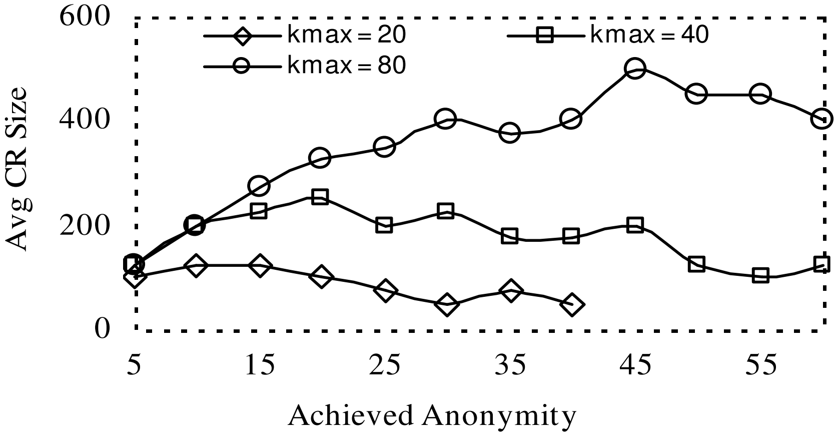
Besides ensuring the reciprocity condition, our proposed approach intends to provide location privacy solution satisfying *k*-anonymity along with *s*-proximity using a location anonymizer. A thorough survey of literature reveals that lots of works have been done to deal with location privacy but none has proposed the inclusion of parameter like *s*-proximity. Existing approaches in achieving anonymity for the LBS services [Mokbel06, Gedik05, Langheinrich02, Ghinita07] have ignored the fact that static information is required during the service access. In [Ghinita07] it is shown that the knowledge of the attacker can be used to perform the re-identification attack. The selection of quasi identifiers [Kalnis07, Sweeney02] from contextual information can place the individual privacy at serious risk [Talukder08]. Most of the existing approaches in the literature fail to protect identity inference caused by the attacks we have shown. In [Machanavajjhala06, Hashem07] it is shown how individual's identity can be inferred even from a *k*-anonymized data set. They proved that *k*-anonymity is not a sufficient measure against re-identification attacks. To protect this, the notion of *l*-diversity is proposed in [Machanavajjhala06] whereas [Hashem07] shows the weakness of *l*-diversity and eliminates those by introducing the concept of t-closeness. These works are similar to our approach. They showed the identity inference attacks in scenario of micro-data publishing whereas we have formulated the attacks in case of using location based services. The generic view of the problem was addressed while considering the disclosure of a number of contexts and static information involved during the service access [Talukder08]. They presented the concept of contexts of a query and related static user information from a theoretical viewpoint. We provide a practical solution of the problem.

Chapter 6: Experimental Evaluation

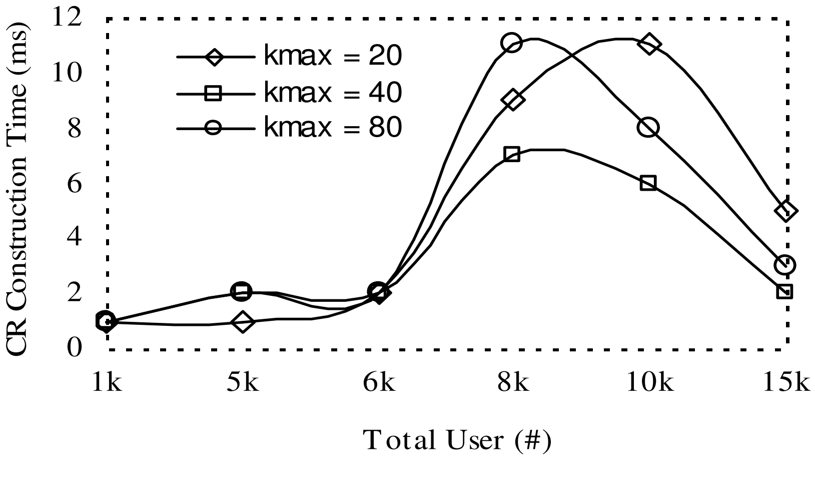
We set the simulation with North-American data set [Hoh05] and used our proposed algorithms. Simulation was developed with Java 1.6 and it was conducted on a machine with hardware configuration Intel M processor 1.7 GHz, 1.5 GB Memory and Windows Vista as OS. First we evaluate our system and then we compare with IC [Gruteser03], HC [Ghinita07], Casper [Mokbel06], NNC [Kalnis07] and PrivacyGrid [Bamba08].

There are two aspects of performance of a location privacy model. These are: level of privacy, which is measured by achieved anonymity level, and quality of service (QOS). QOS depends on two factors. At client side it is related to the processing time which depends on the number of POI returned to the client and this eventually is determined by the CR size. QOS also depends on the CR construction time at the location anonymizer. We have conducted extensive analysis on how SafeGrid performs in terms of these parameters.

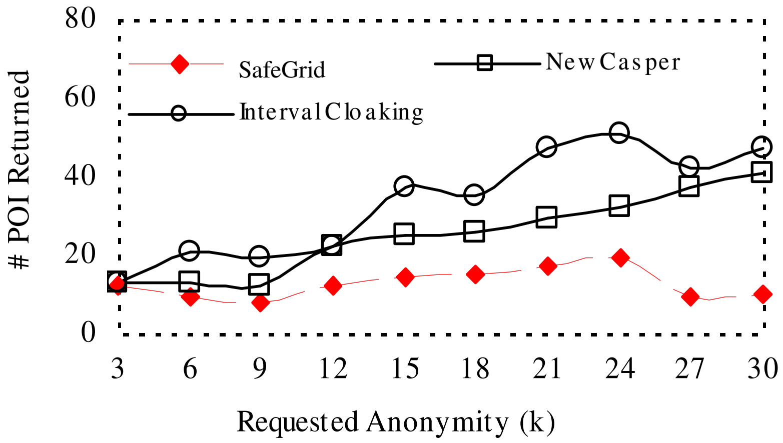
Average size of cloaked region is an important performance metric because processing time at the client side directly depends on it. The way SafeGrid constructs anonymization sets makes higher anonymity requested in a cell significant. Once a cell is taken in a CR, it is locked and users with lower *k* requirement are automatically put into the corresponding AS. That is why kmax (maximum requested k by any user) is a significant parameter and its impact on Avg CR size is depicted in Figure 6.1(a). As kmax increases, Avg CR size gets larger. This means that for someone with abnormally high *k* requirement, others' QOS degrades a bit at the gain of increased privacy. As the total number of users increases, more time is needed to construct CR (see Figure 6.1(b)). This computation is performed at the location anonymizer, so it has little impact on the client side performance. We have compared performance of SafeGrid with the similar best known approaches in literature like IC [Gruteser03], HC [Ghinita07], New Casper [Mokbel06], NNC [Kalnis07] and PrivacyGrid [Bamba08]. All these approaches are similar to SafeGrid as they involve location anonymizer for AS formation and CR construction. Among them PrivacyGrid is the only grid based approach but it does not meet reciprocity condition. Other approaches try to create optimal CR by reducing its size and construction time. Only HC [Ghinita07] meets reciprocity condition.



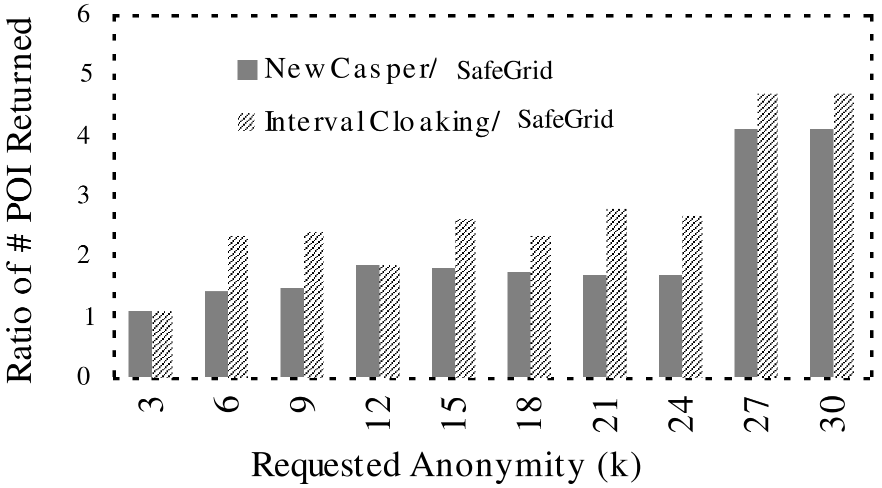
(a) Avg CR Size vs. Achieved Anon. [Total User=10000 GridRes=5.0]



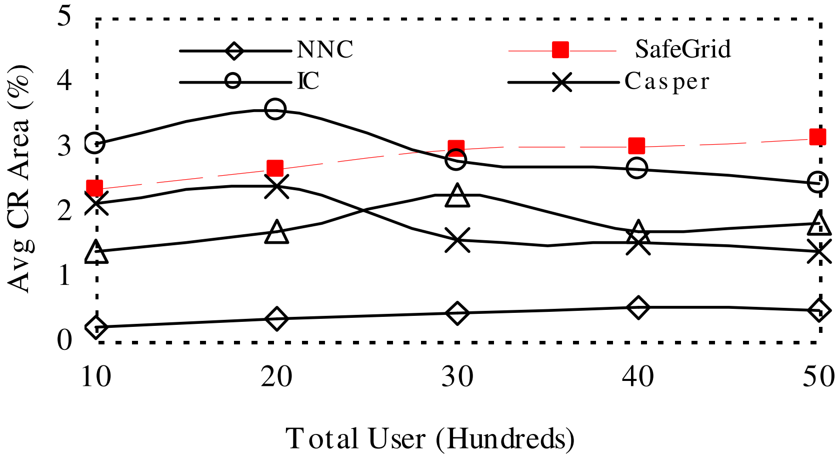
(b) CR Construction Time vs. Total User [GridRes=5.0]



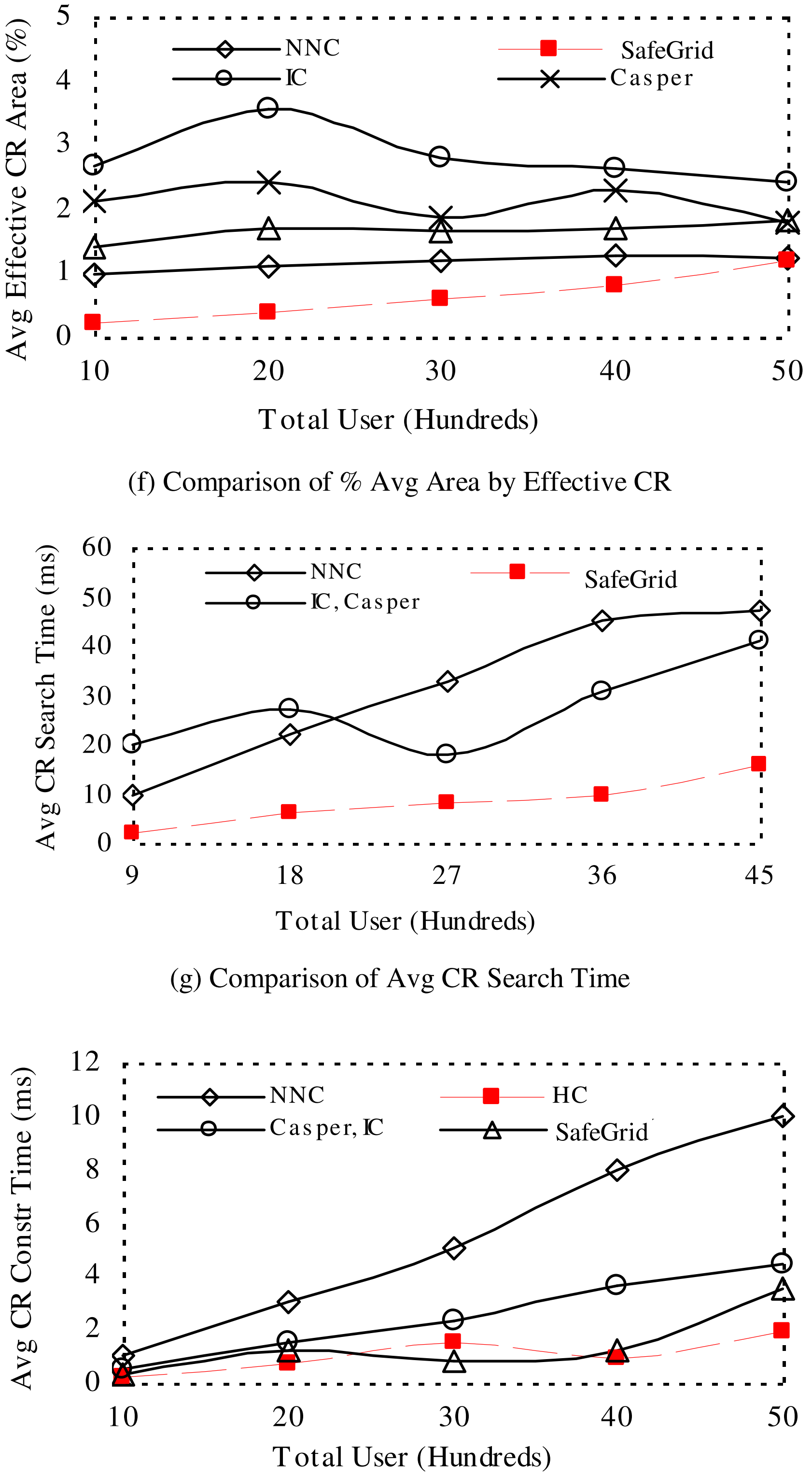
(c) SafeGrid vs. New Casper & IC



(d) SafeGrid vs. New Casper & IC



(e) Comparison of % Avg Area by CR



(f) Comparison of % Avg Area by Effective CR

(g) Comparison of Avg CR Search Time

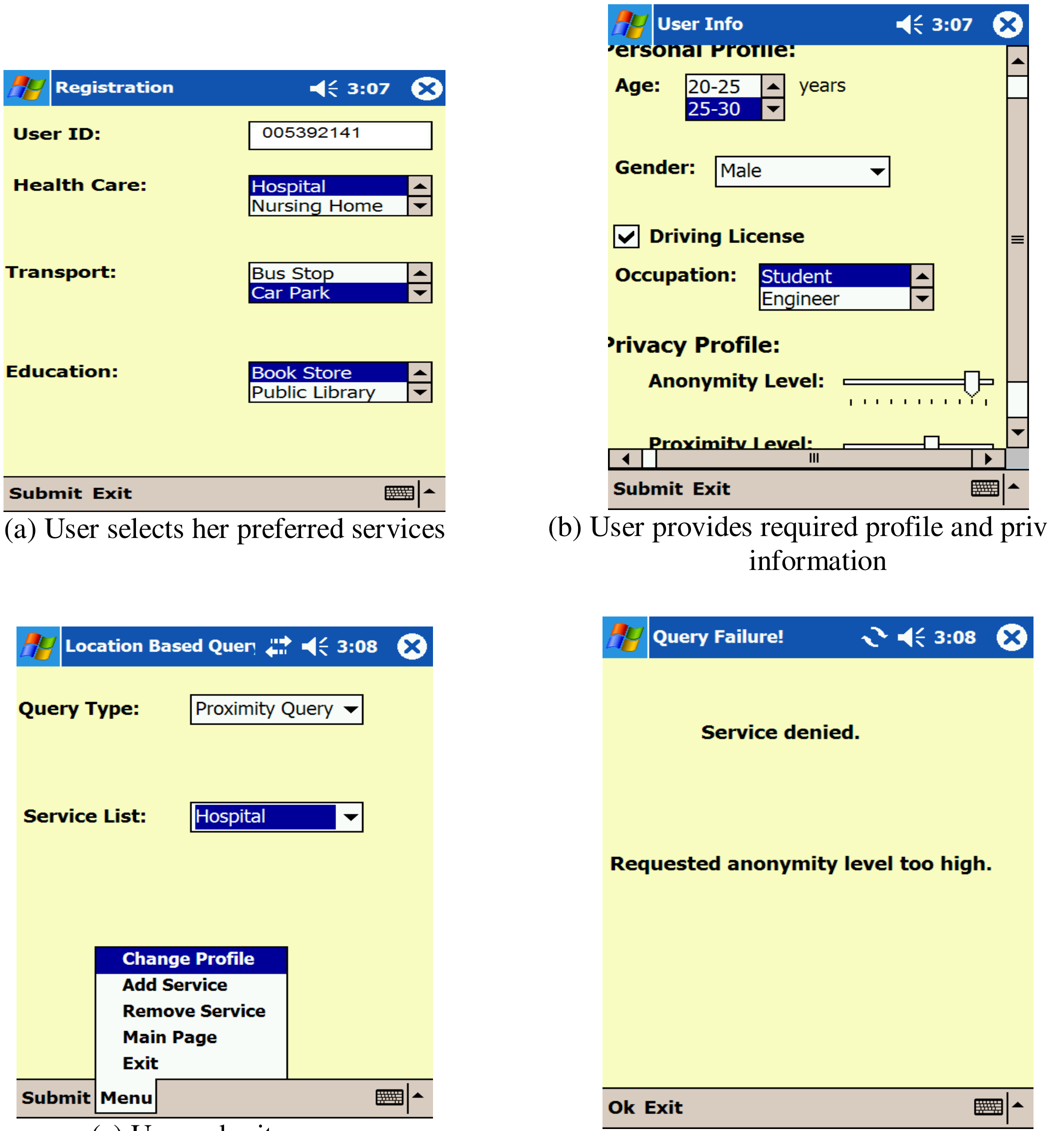
(h) Comparison of Avg CR Construction Time

Figure 6.1. SafeGrid Performance

Figure 6.1(c) & (d) show clear improvement of SafeGrid over IC and New Casper in terms of number of POI returned to the client. In SafeGrid, size of POI list returned to the query requester remains almost constant as LA finally returns only the POIs locating inside the query requester's cell. This is how our approach resolves the tradeoff between privacy and QOS (as claimed in section V(B)). Figure 6.1(e) exhibits that apparently SafeGrid generates larger CRs compared to the peer solutions. However, if we consider the size of effective CR, as shown in Fig 6.1(f), our approach gains much improvement and even it is comparable to NNC [Kalnis07]. Moreover, from Figure 6.1(h) we find that NNC takes very high CR construction time. Here we also find that SafeGrid reduces the CR construction time and CR search time on average [see Fig 6.1(g) (h)]. Other approaches perform poorly as they construct CR every time a new query is issued. These data also demonstrate that SafeGrid is more scalable than others as increased number of users cannot impact its performance.

We have implemented a prototype version of our proposed system. The modules of context aware location anonymizer were developed on a machine with hardware configuration Intel Processor 1.7 Ghz, 1.5 GB Memory and Windows Vista as OS. We have deployed an application that uses, on client side, a Dell Axim X50v pocket PC (Processor type is Intel 624 MHz Xscale, ROM is 128MB Flash). The underlying OS is WinCE and the implementation language is C# on .NET Compact framework.

The spatial data used in the evaluation were taken from North American data set [Hoh05] consisted of 15K points which were used by clients and the LA as user points in 2D space. Figure 4 depicts how the client module works. The initialization step consisting of the user registration tasks are shown in Figure 6.2 (a) (b). The subsequent figures display how a registered user submits location based query to the LA and gets reply subsequently.



Registration 3:07

User ID: 005392141

Health Care: Hospital, Nursing Home

Transport: Bus Stop, Car Park

Education: Book Store, Public Library

(a) User selects her preferred services

User Info 3:07

ersonal Profile:

Age: 20-25, 25-30 years

Gender: Male

☑ Driving License

Occupation: Student, Engineer

rivacy Profile:

Anonymity Level:

Proximity Level:

(b) User provides required profile and privacy information

Location Based Quer 3:08

Query Type: Proximity Query

Service List: Hospital

(c) User submits query

Query Failure! 3:08

Service denied.

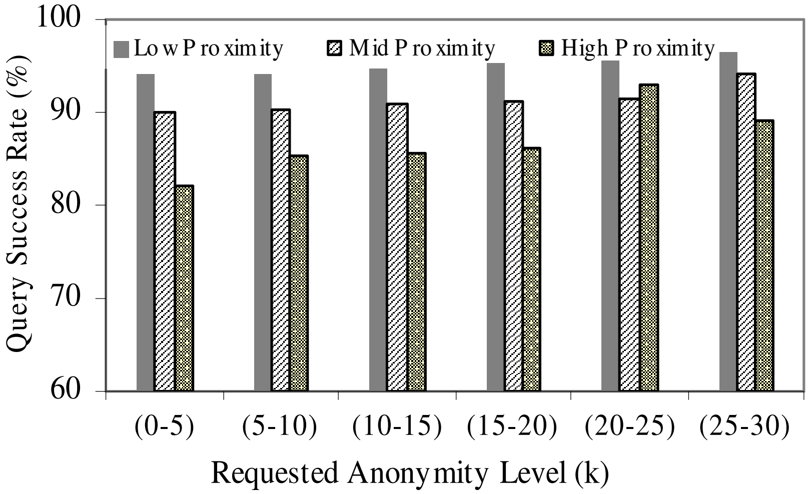
Requested anonymity level too high.

(d) Reply from LA with failure notification

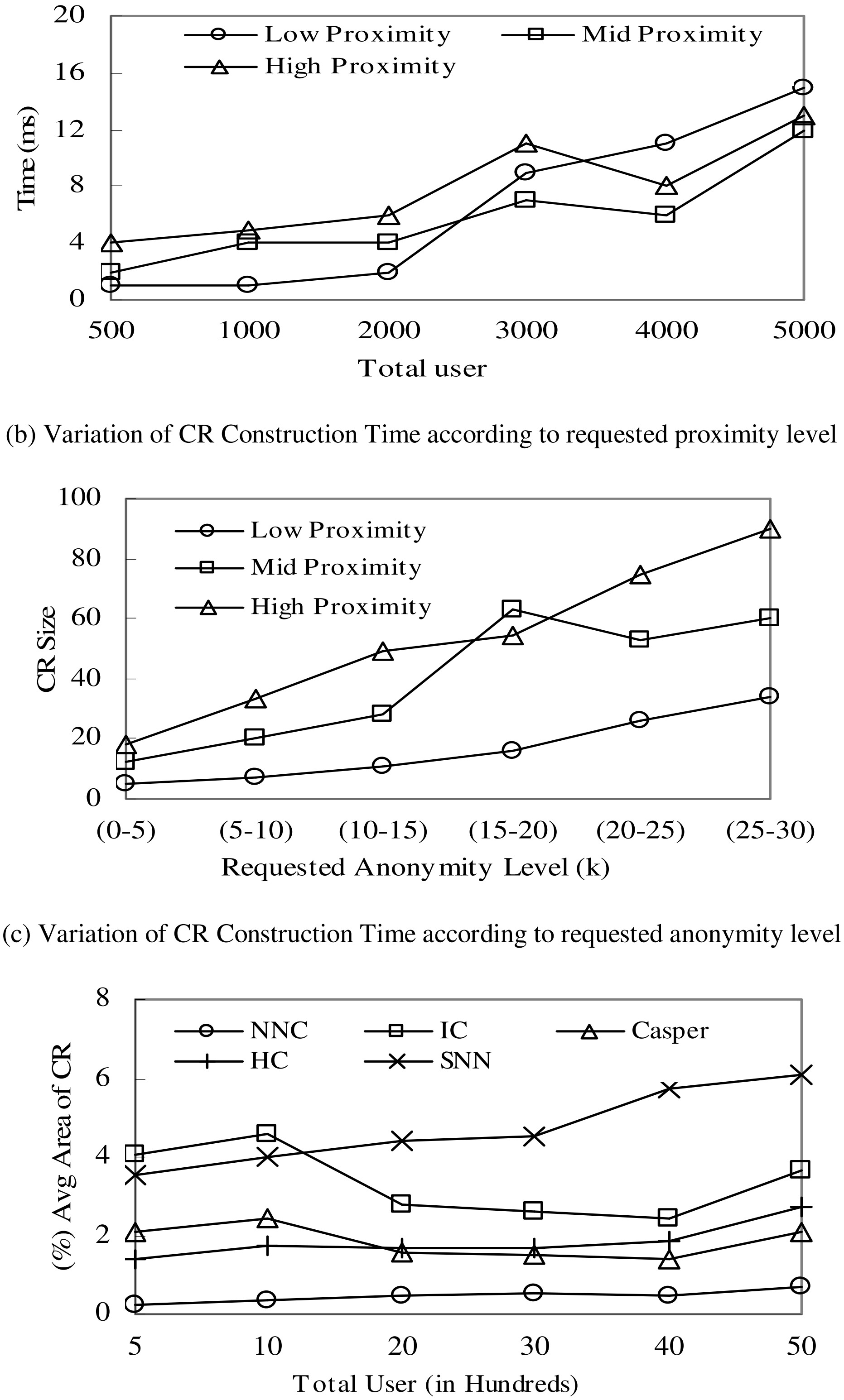
Figure 6.2 Prototype Implementation (User Interface Module)

We have evaluated performance of our final system, which implements both *k*-anonymity and *s*-proximity, and the findings are summarized in Figure 6.3. Performance of the system was measured in terms of the metrics: Query Success Rate, CR Construction Time and CR Size (absolute /relative). The percentage of time a user was provided with her required service was denoted by Query Success Rate. Other two metrics measured the time required for constructing a cloaked region and the size of the constructed CR. As Figure 6.3 (a) depicts, we achieved overall high success rate though it reduced a bit with higher proximity requirement. The graphs in Figure 6.3 (b) (C) demonstrate that the construction time and size of a cloaked region increases with higher proximity level. These parameters also show higher values for increased number of subscribers. Performance of our prototype implementation was compared with couple of other existing systems (IC [Gruteser03], HC [Ghinita07], Casper [Mokbel06], NNC [Kalnis07]) and it is found that our approach yields cloaked region with a bit large size (shown in Figure 6.3 (d)) which is quite acceptable considering the enhanced level of privacy it offers compared to the existing frameworks.

From the experimental facts it is evident that our proposed framework with a context aware location anonymizer is really feasible to be implemented in real world LBS system. Performance of the system is acceptable as compared to existing systems. The system implements both *s*-proximity and *k*-anonymity which is a novel approach capable of safeguarding the attacks presented in this paper.



(a) Query Success Rate



(b) Variation of CR Construction Time according to requested proximity level

(c) Variation of CR Construction Time according to requested anonymity level

(d) Comparison of our approach with existing approaches in terms of relative CR size.

Figure 6.3 Evaluation Results

Chapter 7: Conclusion and Future Work

Location-based Services (LBS) have become very popular in these days due to the increasing trend of high-end smart-phone usage. Yet, this popularity is sometimes diminished by the concern of end users' location privacy violation. The concern stems from the disclosure of location information by the end user or request issuer of LBS when he or she submits the location-dependent query or spatial query to the service provider. Various methods found in the literature protect location privacy by hiding or obfuscating exact location with several other users' location in the system from the service provider. However, none of the methods is still able to provide location based services with full immunity from privacy attacks. The concluding note of this thesis includes a summary of overall contributions and future directions for the research.

7.1 Contribution of the Thesis

In this thesis, we introduce several novel attacks causing serious privacy concern to LBS. In this sort of attacks, the exact location of the requester can be inferred by the adversary through obtaining cloaking regions (CR) that are shrunk or extended in subsequent queries. The remedy to this problem is to retain cloaking regions that contain the same set of users over a period of time (reciprocity condition). Most of the existing approaches demonstrate a fundamental flaw by considering only a static snapshot of users during evaluation. Thus, any modification to the data structure due to mobile user location update turns out to be very costly. Our proposed approach is principled on developing disjoint sets of users dynamically over time in order to share the common CRs.

In addition, we have proposed a location privacy solution with a trusted third party *(c-LA)* equipped with additional functionalities which are accomplished prior to AS construction. The algorithms presented in this thesis aim to construct an AS which meets *k*-anonymity and *s*-proximity. The novel privacy feature *s*-proximity is proposed as a solution of couple of plausible attacks applicable against most of the existing approaches. We have implemented a prototype version of the system. Evaluation results demonstrate the feasibility of our proposed approach along with its performance measures. The experimental study supports our claim that SafeGrid is efficient and suitable for the mobile environment compared to other approaches. In future, we plan to build real world LBS applications using the SafeGrid framework.

7.2 Broader Impact of the Research

The major contribution of the research will be an attack-resistant privacy protection technique for Location-based Services running on resource constrained handheld devices that use a Location Anonymizer. The application designers can benefit through adopting this approach in their designs and tailoring it to their specific needs by regulating various system parameters. The research will have a wide range of impacts on the traffic monitoring system, weather forecast system, health care industry (personal tracking system), corporate office premise (internal messenger system), school campus (buddy locator, campus map), customer support (tracking application to ensure quality of service), advertisement, marketing, tourism industry and many more.

SafeGrid is going to add new dimension to location privacy research in LBS where typically the networks are comprised of resource-constrained mobile devices. The researchers and the students who work in this area will be able to look into and further contribute to the research results of the thesis, which will be made publicly available. The data sets used and experimental results to evaluate the protection model in the thesis can be used in promoting research in this area. Finally, the research will benefit the research community to advance in the location privacy research with the new dimensions of thinking, and provide the handheld device programmers a unique opportunity to model privacy in their applications, facilitating the onward journey of pervasive computing.

The proposed location privacy framework is generic and flexible so it can be easily incorporated into existing and new systems that attempt to better protect users' identities and their private information, without major modifications to the system and network infrastructure. The proposed approach challenges the traditional belief of the zero-sum tradeoffs between privacy and QoS, which can result in a major paradigm shift. The research outcome will effectively disseminate this new paradigm to the next generation of workforce in the related fields. It will help to the research community to develop new ideas and solutions when handling the privacy and security issues in location based systems and other distributed systems.

7.3 Future Work

The proposed framework SafeGrid, indeed, opens up new avenues for research in location privacy in LBS. Enforcement of reciprocity requirement and identity inference protection are the most interesting aspects of our approach, which the existing approaches have repeatedly ignored in most cases. Some of the future research scopes of SafeGrid and LBS privacy are discussed here.

The performance of SafeGrid can be analyzed by investigating the dynamic reorganization of data structure on user movement with trees of other orders, such as ternary trees. It is important to check how it performs in a dynamic environment with varying speeds of users in a simulated environment. In addition, we plan to compare its performance with other interesting choices of greedy heuristic that can determine disjoint anon sets of neighboring users.

The industry is constantly aiming at taking handheld devices like PDAs, cell phones, and smart phones to the next frontier of technology by accommodating more processing and storage capability. Battery power, however, stands as the bottleneck to long hours of continuous operation for these devices. With faster new gadgets and the requirements for trusting a third party LA, researchers have become very keen on investigating LBS models that get rid of LA [Chow06, Hashem07, Ghinita06,Ghinita08]. The peer-to -peer models [Chow06, Hashem07, Ghinita06] require that the peers establish trust relationships among themselves and share exact location information. In practice, this principle may even raise more privacy concerns. In the approach depicted by Ghinita et. al [Ghinita08], the user retrieves a list of POIs through Private information retrieval method [Chor95] from the LBS. In this way, LBS doesn't know what information has been requested, and sends the data structure on selected POIs (stored as Voronoi Tessellation with POIs) to the user prior to issuing a query. The user selects a number of POIs from the list and sends back to the server in order to determine the CR, which doesn't involve any anon level. The issues with this approach are that the density control of the POIs is arbitrary at the provider's end, and the additional overhead of filtering the result set is now the query issuer's task. Future direction of privacy research in LBS is to find robust privacy measure, suitable data structures and secured protocol to support the very nature of resource-constrained devices.

BIBLIOGRAPHY

[Bamba08] Bamba, B., Liu, L., Pesti, P., Wang, T. (2008). *Supporting Anonymous Location Queries in Mobile Environments with PrivacyGrid*, Proc of WWW, 2008, pp. 237-246.

[Bettini07] Bettini, C., Mascetti, S., Wang, X.S., Jajodia, S. (2007). *Anonymity in Location-Based Services: Towards a General Framework*, Proc. of MDM, 2007, pp. 69-76.

[BuddyBeacon] Buddy Beacon iPhone Application:http://www.where.com/buddybeacon/

[Chow06] Chow, C.Y., Mokbel, M.F., Liu, X. (2006). *A peer-to-peer spatial cloaking algorithm for anonymous location-based services*, Proc. of ACM Symp. on Advances in GIS, 2006, pp. 171–178.

[Cuellar02] Cuellar, J., Morris, J.B., Mulligan, D. (2002). *Geopriv Requirements*, Internet draft, Nov. 2002.

[Dey01] Dey, A. K. (2001). *Understanding and Using Context*, in Personal and Ubiquitous Computing 5(1), pp 4-7 (2001).

[Duckham05] Duckham, M., Kulik, L. (2005). *A formal model of obfuscation and negotiation for location privacy*, Proc. of 3rd International Conference on Pervasive Computing, Springer-Verlag, 2005, pp. 152-170.

[EarthComber] Earth Comber: Finding Point of interests without GPS: http://www.apple.com/webapps/searchtools/earthcomberusa.html

[FindChild] FindYourChild http://www.findyourchild.net/

[Gedik05] Gedik, B., Liu, L. (2005). *A Customizable k-Anonymity Model for Protecting Location Privacy*, Proc. of ICDCS, 2005, pp. 620-629.

[Gedik05] Gedik, B., Liu, L. (2005). *Location-Privacy in Mobile Systems: A Personalized Anonymization Model*, Proc. of ICDCS, 2005, pp. 620-629.

[Gedik08] Gedik, B., Liu, L. (2008). *Protecting Location Privacy with Personalized k-Anonymity*, IEEE Trans. Mobile Computing, vol. 7, no. 1, 2008, pp. 1-18.

[Ghinita07] Ghinita, G., Kalnis, P., Skiadopoulos, S. (2007). *PRIVÉ: Anonymous Location-Based Queries in Distributed Mobile Systems*, Proc of WWW, 2007, pp. 371-380.

[Ghinita08] Ghinita, G., Kalnis, P., khosgozraran, A., Shahabi, C., Tan, K. (2008). *Private Queries in Location Based Services: Anonymizers Are Not Necessary*, Proc. of SIGMOD, 2008, pp. 121-132.

[Gruteser03] Gruteser, M., Grunwald, D. (2003). *Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking*, Proc. of MobiSys, 2003, pp. 31-42.

[Gunter04] A. C. Gunter, M. J. May, S. G. Stubblebine (2004). *A Formal Privacy System and its Application to Location Based Service,* Paper presented at the Workshop on Privacy Enhancing Technologies (2004) , Toronto, Canada.

[Hasan09] Hasan, C.S., Ahamed, S.I., Tanviruzzaman, M. (2009). *A Privacy Enhancing Approach for Identity Inference Protection in Location-Based Services*, Proceedings of the 33rd Annual International Computer and Software Applications Conference (COMPSAC), 2009, pp. 1-10.

[Hashem07] Hashem, T., Kulik, L. (2007). *Safeguarding Location Privacy in Wireless Ad-hoc Networks*, Proc. of Ubicomp, 2007, pp. 372-390.

[Hengartner06] Hengartner, U., Steenkiste, P. (2006). *Avoiding Privacy Violations by Context-Sensitive Services*, in Fourth Annual IEEE International Conference on Pervasive Computing and Communications (Percom), 2006.

[Hoh05] Hoh, B., Gruteser, M. (2005). *Protecting Location Privacy Through Path Confusion*, Proc. of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, 2005, pp. 194 – 205.

[Kalnis07] Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D. (2007). *Preventing Location-Based Identity Inference in Anonymous Spatial Queries,* IEEE Transactions on Knowledge and Data Engineering, v. 19, no. 12, December 2007, p. 1719-1733.

[Kapadia08] Kapadia, A., Triandopoulos, N., Cornelius, C., Peebles, D., Kotz, D. (2008). *AnonySense: Opportunistic and Privacy-Preserving Context Collection*, In Proc of Pervasive 2008

[Langheinrich02] Langheinrich, M. (2002). *A Privacy Awareness System for Ubiquitous Computing Environments*, Proc. of Ubicomp, Springer-Verlag, 2002, pp. 237-245.

[Li07] Li, N., Li, T., Venkatsubramanian, S. (2007). *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity*, Proc. of ICDE, 2007, pp. 106-115.

[Machanavajjhala06] Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M. (2006). *l-Diversity: Privacy Beyond k-Anonymity*, Proc. of ICDE, 2006, pp. 24.

[Mascetti09] Mascetti, S., Bettini, C., Wang, X.S., Freni, D., Jajodia, S. (2009) *ProvidentHider: An Algorithm to Preserve Historical k-Anonymity in LBS*, Proc. of 10th Int. Conf. on Mobile Data Management, 2009, pp.172-181.

[Mokbel06] Mokbel, M.F., Chow, C., Aref, W.G. (2006). *The New Casper: Query Processing for Location Services without Compromising Privacy*, Proc. of VLDB, 2006, pp. 763-774.

[Monjur09] Monjur, M., Ahamed, S.I., Hasan, C.S. (2009). *ELALPS: A Framework to Eliminate Location Anonymizer from Location Privacy Systems*, Proceedings of the 33rd Annual International Computer and Software Applications Conference (COMPSAC), 2009, pp. 11-20.

[Palen03] Palen, L., Dourish, P. (2003). *Unpacking "Privacy" for a Networked World*, In Proc of the Conference on Human Factors in Computing Systems (CHI 2003), pp. 129-136.

[PocketFinder] PocketFinder Location based Application: http://pocketfinder.com/

[POLS] Privacy Observant Location System: http://pols.sourceforge.net/

[Robinson04] Robinson, P., Vogt, H., Wagealla, W. (2004). *Some Research challenges in pervasive computing*, Post workshop at the second international conference on pervasive computing, April 18-23, 2004, Vienna, Austria, pp. 1-16

[Satya96] Satyanarayanan, M. (1996). *Fundamental Challenges in Mobile Computing*, Fifteenth ACM Symposium on Principles of Distributed Computing, 1996.

[Schimdt99] Schmidt, A., Beigl, M., Gellersen, H. (1998). *There is more to context than location*, Proceedings of the International Workshop on Interactive Applications of Mobile Computing (IMC98), November 1998, Rostock, Germany

[Sweeney02] Sweeney, L. (2002). *K-anonymity: a model for protecting privacy*, International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002; pp. 557-570. A. K. Dey, “Understanding and Using Context”, Personal and Ubiquitous Computing, 2001, pp. 4-7.

[Talukder08] Talukder, N., Ahamed, S.I. (2008). *FPCS: A Formal Approach for Privacy-Aware Context-Based Services*, Proc. of COMPSAC ,2008. pp. 432 – 439.

[Talukder10] Talukder, N., Ahamed, S.I. (2010). *Preventing Multi-query Attack in Location-based Services*, Proceedings of the Third ACM Conference on Wireless Network Security (WiSec), 2010, New Jersey, USA, pp. 25-36.

[Tang06] Tang, K.P., Fogarty, J., Keyani, P., Hong, J.I. (2006). *Putting People in their Place: An Anonymous and Privacy Sensitive Approach to Collecting Sensed Data in Location-Based Applications*, Proceedings of ACM Conference on Human Factors in Computing Systems (CHI2006), CHI Letters, 2006. 8(1): pp. 93-102

[Want05] Want, R., Pering, T. (2005). *System challenges for ubiquitous & pervasive computing*, in *27th International Conference on Software Engineering (ICSE 2005)*, St. Louis, Missouri, USA, May 15-21, pp. 9-14

[Weiser91] Weiser, M. (1991). *The Computer for the Twenty-First Century*, Scientific American, September 1991, pp. 94-104

[Westin67] Westin, A. F. (1967). *Privacy and Freedom*, New York NY: Atheneum, 1967.

[Xu09] Xu, T., Cai, Y. (2009). *Feeling-based Location Privacy Protection in Location-based Services*, Proc. of ACM CCS, 2009, pp. 348-357.

[Xu10] Xu, T. Cai, Y. (2010). *Location Cloaking for Safety Protection of Ad Hoc Networks*, *IEEE INFOCOM'10*, Rio de Janeiro, Brazil, 2010.