TRANSACTIONS ON DATA PRIVACY 2 (2009) 77–99

Towards Achieving Personalized Privacy for Location-Based Services

Nayot Poolsappasit, Indrakshi Ray

Computer Science Department, Colorado State University, Fort Collins, Colorado 80523, USA.

Email: nayot@cs.colostate.edu, iray@cs.colostate.edu

Abstract. With the growth of wireless and mobile technologies, we are witnessing an increase in location-based services (LBSs). Although LBSs provide enhanced functionalities, they open up new vulnerabilities that can be exploited to cause security and privacy breaches. Consequently, location data of individuals used by such services must be adequately protected. Such services will require new models for expressing privacy preferences for location data and mechanisms for enforcing them. We identify the factors on which location privacy depends and propose models for expressing privacy that can be used by LBSs. We discuss the architecture of a system that allows one to specify and enforce location privacy and that can be easily integrated with existing systems providing LBSs. We demonstrate the feasibility of our approach by developing a prototype.

1 Introduction

Recent research on location technology has spawned numerous services, such as, FCC's Enhanced 911, AxisMobile's FriendZone, Verizon's Navigator, Sprint's Family Locator, RIM's Blackberry Service, or Intel's Thing Finder, that reveal location information with a high-degree of spatial precision. Such technology will not only provide enhanced functionality, but will also introduce additional security and privacy concerns. Improper protection of location information may have grave consequences, such as compromising the physical security of users. Thus, location information of individuals subscribing to or using such services must be protected against security and privacy breaches. Models are needed that will allow individuals to express their privacy preferences and technologies are needed to enforce them.

Location-Based Services (LBSs) can be classified into various categories based on their functionality. Examples of LBS applications include navigation (directions, traffic control), information (travel and tourist guides), tracking (people, vehicle or product tracking), emergency (police, ambulance), advertising (advertisement alerts), billing (road tolls), and social networking (locating friends, instant messaging). Controlled disclosure of location information is important for some of these applications. For example, a service like advertisement alert or travel and tourist guides may need to verify legitimate use but may not need other personal information about the user requesting the service. Privacy issues are not important for such applications. Some services may need location information of the requester but not his exact identity. Examples include navigation and road tolling services. Privacy is also not important for such services. Other services, such as, people tracking and social networking, need both the identity of the user as well as his exact location. Controlled disclosure of location information is critical for such applications. We focus on this

category of applications and show how location information of a user can be disclosed such that it respects his privacy.

The notion of privacy varies from one individual to another. One individual may be willing to disclose his location to his co-workers while he is on vacation, whereas another individual may not want to do so. The other issue is the granularity at which end users are willing to disclose their location information. For example, a person may be willing to disclose to his friends that he is in town on a particular day, but may not be willing to reveal his exact whereabouts. The key question in location privacy is who should have access to what location information and under what circumstances. Ideally, we need a model that will allow different users to express their location privacy preferences and mechanisms for enforcing them.

Developing a generalized model that takes into account the personal privacy preferences of all potential individual users is not feasible. Consequently, we have identified some factors, namely, identity or role, usage, time and location, that are important for location privacy. Identity or role specifies the requester who is requesting the location information, usage identifies the purpose of this request, time denotes the time of the request and location gives the location of the requested object. We require a user to express his location privacy in terms of these factors.

We propose three different models that use these factors for expressing privacy preferences. The models differ with respect to the computation requirements, and the granularity with which privacy preferences can be expressed. We show how to compute the level of disclosure for a given query using the privacy preference of a user and illustrate how it can be related to existing privacy models, such as k-anonymity.

We also discuss implementation issues pertaining to our model. We present our system architecture for enforcing location privacy and show how it can be integrated with existing systems supporting LBSs. We provide an analysis of the attacks that are possible in our system and what protection measures are needed to protect against those attacks. We also discuss how typical location-based queries can be processed in our system, and develop a prototype to demonstrate the feasibility of our model.

The rest of the paper is organized as follows. Section 2 briefly enumerates the related work in this area. Section 3 describes the problem that we are trying to solve. Section 4 identifies the factors that are important to location privacy and proposes techniques for quantifying them. Section 5 discusses how to compute the information disclosure and the privacy levels. Section 6 illustrates our proposed system architecture. Section 7 briefly describes the modules that we have implemented. Section 8 discusses how some typical location-based queries are handled by our system and shows some simulation results. Section 9 analyzes the security threats that are present and how we can mitigate them. Section 10 concludes the paper and mentions some future works.

2 Related Work

IETF Geographic Location Privacy (GEO-PRIV) working group [7] addresses privacy and security issues pertaining to transferring and storing location information. They focus on the design of communication protocol to ensure confidentiality and integrity of data.

Researchers also proposed access control models where access is contingent upon the location of users and objects [1, 2, 8, 12, 15, 20, 19, 21]. However, most of these works do not discuss how location information must be protected to prevent privacy breaches.

Several approaches have been proposed for achieving location privacy. Some of these

approaches use the notion of *k*-anonymity where the location of a mobile user is indistinguishable among *k* users [3, 11, 16]. Gruteser and Grunwald [11] propose a spatio-temporal cloaking algorithm that allows the user's location to be indistinguishable from k people. The mobile nodes communicate with services using a trusted anonymity server which is responsible for removing any identifiers from the response and changes position data using the proposed cloaking algorithms. The problem is that if some nodes are malicious, they can collude and compromise the privacy of a targeted user. The Clique-Cloak algorithm [9] takes a similar approach as [11]. It builds a clique graph from a set of all subscribed users which is used to decide whether some users share the cloaked spatial area. Due to the computation overhead of the clique graph, this approach does not scale very well. It becomes especially problematic for users that require a high value of *k* for *k*-anonymity.

Mokbel et al. [16] propose the Casper framework that allows users to use location-based services without disclosing their location information. It consists of two parts: location anonymizer and privacy-aware query processor. Location anonymizer blurs the user's exact location and responds with a cloaked region. Privacy aware query processor returns the location query results corresponding to the cloaked region which must then be refined to get the exact response at an additional computation cost.

Chi-Yin Chow et. al. [3] propose a P2P spatial cloaking approach that does not require the use of a trusted third party. The proposed scheme exploits anonymous peer-to-peer searching to construct the cloaked region such that user cannot be distinguished from *k* other users in the cloaked area. Then, the user selects an agent among the *k* entities who is responsible for forwarding the user's query to the service provider. This approach has some disadvantages. First, forming the group and selecting the agent may be challenging because not all mobile devices subscribe to the same service provider. Moreover, it assumes that the peers are trusted enough and do not compromise the privacy of individuals.

Ghinita et al. [10] propose new location privacy framework based on the theory of private information retrieval. The key idea is the user sends the generalized query to LBS, LBS replies with a set of results from which the requester retrieves the actual result. Since the LBS is unaware of the exact location of the user, location privacy is protected. The advantage of this approach is that it does not require LBS or any third party to be trusted. However, it does not shed any light on how users can generalize their queries when they do not have knowledge about how the data is organized. The approach also assumes that mobile devices have enough computation capabilities to filter out the useless responses.

Kido et. al. [14] propose achieving location privacy using dummy locations. The main idea is that the user sends a set of false location called dummies along with true location to the Server. The location server processes all requests and sends all answers back to the subscriber. The client then picks only one answer it desires from the candidate list. Clearly, the disadvantage of this approach is that the server wastes a lot of computation resource in processing false queries and the adversary may detect the true location from observing the request history.

Ren et al. [22] propose a scheme to provide anonymous but authorized communication between mobile users and service providers in pervasive computing environments. The scheme uses an authenticator, which is a trusted third party, to provide mutual authentication between users and services to authenticate and authorize the service request. The user obtains a certificate from the authenticator that allows him to obtain the service while remaining anonymous. The proposed scheme is resistant to eavesdropping, impersonation and service spoofing attacks. User privacy can be breached only if both authenticator and service provider are compromised (compromise authenticator to breach user's identity and compromise service provider to reveal the transaction).

Jiang et al. [13] propose a technique based on using pseudonyms to protect the location privacy in wireless network environment. The approach achieves location privacy by frequently changing the device's pseudonym and introducing a silent period to suppress device's transmission signal. Doing so guarantees that the user uses different pseudonym on each communication. Hence, it lowers the chance of being attacked. Note that, the user cannot avail of any service during the silent period. The approach may be good for off-line services (e.g. point of interest or traveling plan service) but not very suitable for event-driven and real-time services.

Cranor et al. proposed Platform for Privacy Preferences (P3P) [5, 6, 25] for specifying privacy policies and preferences with respect to web services. Our expression for location privacy was inspired by P3P, although our specification for location privacy is simpler as it involves fewer factors.

We were also influenced by several social studies [4, 17, 23] pertaining to the disclosure of private information. Palen et al. [17] found that the privacy management is a dynamic response to circumstances rather than a static enforcement of rules. They emphasized that the social and institutional setting must be considered in developing privacy-aware technology. In response to [17], Consolvo [4], and Smith [23] worked on issues related to the disclosure of location information. Their results conclude that people reveal their location based on *who* is requesting, *why* they want to know the location, *when* and *where* the policy owner is, and *how* the policy owner feels about the requester at the time of request. We have tried to incorporate some of these results in our proposed privacy preference model.

Our work is closest to Snekkenes's location privacy model [24]. Snekkenes identifies five components that play a major role in location privacy: requester, object, usage, time, and velocity and propose a lattice-based approach for location privacy. Since the complete lattice containing information corresponding to all the different factors is very large, the author proposes using a sparse lattice. This sparse lattice covers circumstances that the policy owner anticipated. To handle unexpected situations, the unforeseen scenario is matched with the predefined circumstances that have at least one element in common. The paper does not specify the minimum requirements needed to build the lattice. Without this requirement, the initial information may be too sparse and have inadequate information to determine the preference in certain circumstances. This motivated us to develop a privacy preference model with primitive requirements so that the full policy can be generated from a minimum but adequate set of information [18]. The current work augments the earlier paper by showing how the level of information disclosure can be tied to existing notions of privacy, such as *k*-anonymity, proposing a system architecture for privacy-preserving location-based services, and developing a prototype that allows clients to specify privacy policies and servers to enforce them.

3 Problem Statement

LBSs handle many different types of queries. Examples include "Find the gas stations within a radius of 2 miles", "Where is Smith?", or "Notify subscribers about the traffic on the street they are entering". Location-based queries have different entities associated with them. *Requester* is the entity who issues the location-based query. *Requested object* is the object or entity whose location is being queried. *Service provider* is responsible for providing services to customers. *Subscriber* is the customer who subscribes to services provided by the service provider. *Location provider* is the entity that computes the spatial information and is responsible for respecting the privacy of the location of the requested object. *Policy*

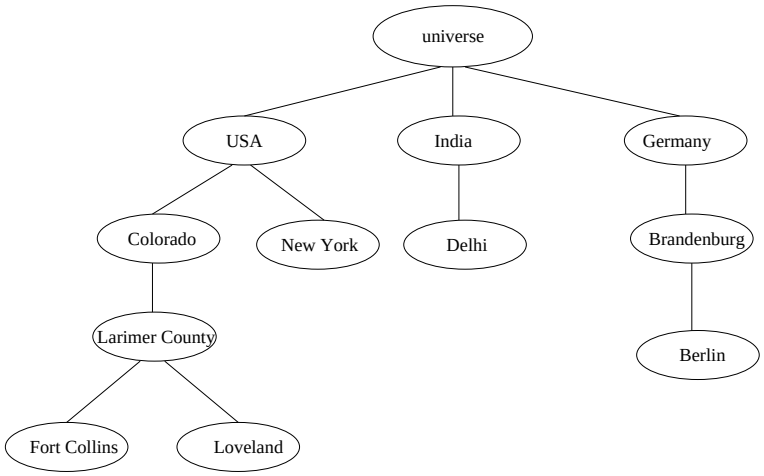

Figure 1: Example of a Location Hierarchy

owner is the entity who decides the location privacy of the requested object.

In this paper, we focus our attention to the queries where the requested object is a user or a device belonging to some user. Note that, for such queries, the location information must be disclosed in a controlled manner to protect the privacy and security of the individuals. The location provider should respect the privacy of the individual owner and provide information to the requester. It is the onus of the policy owner to specify what location information can be revealed to whom and under what circumstances. The factors that influence the willingness of an user to reveal his location information constitute the context of the query.

The response to a location query is the location information. Instead of giving the location in terms of physical coordinates, the system will respond with logical locations. Logical locations are symbolic names associated with physical locations. Examples of logical locations are USA, Colorado, and Fort Collins. We assume that the system has an efficient mechanism for manipulating location data and translating physical locations to logical locations and vice-versa. The logical locations are organized in a hierarchical structure as shown in Figure 1. The nodes represent the different locations. The root of the hierarchy is the location *universe* which contains all other locations. If a node N_i appears higher up in the hierarchy and is connected to node N_j that appears lower in the hierarchy, we say that node N_i contains node N_j and is denoted by $N_j \subseteq N_i$. The hierarchical structure helps determine the location granularity. A user, when specifying his privacy preference, can choose the level of granularity at which he wishes to respond to the query.

The policy owner must provide his location privacy preferences. Location privacy depends on several factors (described in details in Section 4). These factors form the query context. The query context determines the location information that can be revealed to the user. Corresponding to the query context, we store information that specifies the details about location disclosure.

Definition 1. [Location Privacy Preference] The policy owner specifies the privacy preference as a set of tuples of the form $\langle c, loc_c \rangle$, where c is the context of the query and loc_c is the location that is revealed in response to the query.

The response to a location query is said to be correct and privacy preserving if it satisfies several conditions. First, the actual most specific location of the object should be contained in the location that is returned in response to the query. Second, the location that is returned in response to the query should satisfy the location granularity and details that are specified in the privacy preference.

Definition 2. [Privacy Preserving Location Response] Let the context associated with the given query be c and loc_c is the location information associated with context c in the policy owner's privacy preference. Let loc_o be the most specific actual location of the requested object and loc_r be the response that is returned to the user. The location information returned to the user, loc_r, is said to be a *correct privacy preserving response* if $loc_o \subseteq loc_r$ and $loc_c \subseteq loc_r$.

4 Factors Influencing Location Privacy

In order to enforce location privacy, one needs to understand the factors that influence the willingness of an user to reveal his location information. First, the requester's identity or role plays an important part. An user may be willing to reveal his location information to his spouse but may not be willing to do so to strangers. Second, the usage information may also play a role for location privacy. An user may be willing to disclose his location information to volunteers during emergency operations, but may not do so otherwise. Third, the time when the information is requested also plays an important role. A person may reveal his location information to co-workers during his office hours, but may not do so during vacation. Fourth, location itself plays an important role in location privacy. A person may not be willing to reveal his location information when he is in the hospital undergoing some private treatment, but may reveal his location information when he is in the theater.

What makes location privacy a complex problem is the fact that the factors mentioned above are really not independent. Instead, location privacy depends on the combination of these factors. For example, a person may be willing to reveal his location information to his co-workers when he is in the office during the working day, whereas, he may be unwilling to disclose his location information to his spouse when he is in the bar at midnight enjoying with his friends. The combination of these different factors form the context of the location-based query. The response provided by the user depends upon the context of the query.

Definition 3. The context formalizes the scenario under which a location query has been placed. The context of location query l is specified by the tuple $< I_l, U_l, T_l, L_l >$ where I_l represents the identity or role of the requester, U_l denotes the usage requirement of the requester, T_l specifies the time when the query is placed, and L_l is the location of the requested object. The individual entities in the context corresponding to identity, usage, time and location are referred to as context elements.

4.1 Representing the Factors

Since the context of each query has to be matched against the privacy preference of the user, we need a mechanism to represent each factor. For example, one may choose to represent the possible values for the identity factor as a set of strings. Similarly, the other factors can also be represented as sets of strings. The problem with this approach is that there has to be an exact match between the factors specified in the query context and those that are stored

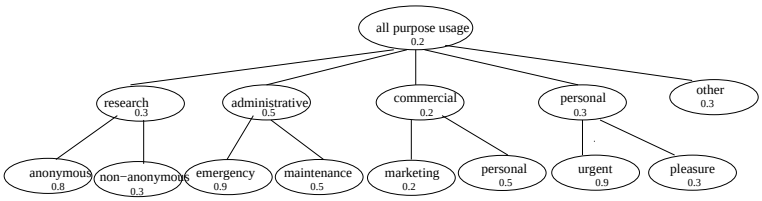

Figure 2: Usage Hierarchy

in the privacy preference profile. In the absence of an exact match, the default response will be returned to the user.

The above problems can be removed to some extent if we quantify each factor in the context. The major advantage of such an approach is that it allows us to extrapolate context values for unknown circumstances. It also makes it easier to calculate the location preference for a given context. In the following, we describe how to assign numerical values to each factor.

4.1.1 Quantifying Requester's Role

Ideally, a person would like to reveal his location information based on the identity of the user. However, such a model will not scale well when there are a very large number of users. Thus, we propose using the role of the requester for determining location privacy. We identify certain important roles for location privacy. Examples include close relatives, close friends, neighbors, co-workers, employers, adversaries, strangers, commercial agents, police, government workers etc. For each of these defined roles, we can adapt Bogardus social distance scale to measure relationship closeness. We assign a value between 0 and 1 for each such role. The value is near to 1 for close relationships and approximates 0 for remote relationships. Certain roles which may not represent close relationships may also be assigned a high value due to the nature of the role. Examples include social worker or law enforcement officer. The reason is that these roles must have access to location information.

4.1.2 Quantifying Usage

The requester must also specify how he is going to use the location information. All potential forms of usage can be organized in the form of a hierarchy. The nodes higher up in the hierarchy signify more general usage than those found lower in the hierarchy. The leaf nodes are assigned values in the range 0 to 1. 0 signifies that the usage is not very important and so information must not be disclosed. 1 signifies legitimate use and must be disclosed. The values for the intermediate node is calculated by taking the minimum value from its children. The process is repeated for the entire hierarchy.

4.1.3 Quantifying Time

The temporal attribute is also an important factor in location privacy. Time can also be represented in the form of a hierarchy. The root of the hierarchy is denoted as *always*. At the next level, we have working hours and non-working hours. Since location privacy is

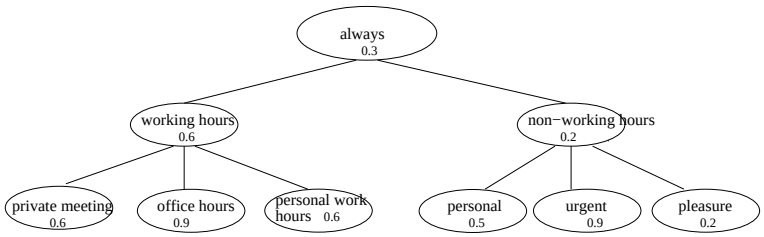

Figure 3: Temporal Hierarchy

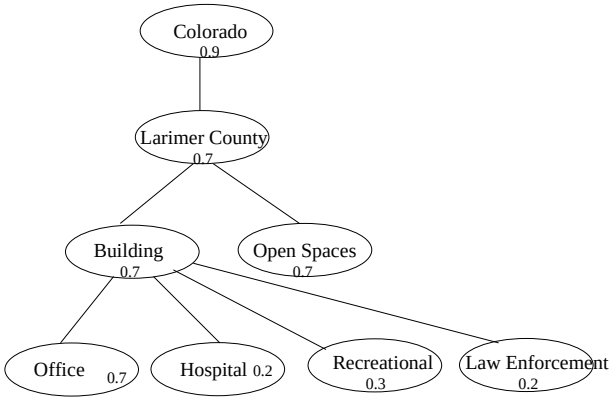

Figure 4: Location Hierarchy

relatively less important during working hours, a value close to 1 is assigned. For non-working hours location privacy may be extremely important and a value close to 0 may be assigned.

4.1.4 Quantifying Location

The propensity to disclose location information may be dependent on location itself. We can organize location in the form of a hierarchy and associate values with it. The values as before range from 0 to 1. Nodes higher up in the hierarchy are assigned a greater value than nodes lower down. This is because a user may be more willing to disclose less granular location information. However, nodes within a level may be assigned different values depending on the sensitivity. For example, the nodes hospital and park have different values associated with them because they differ in sensitivity.

Note that, specifying the hierarchy and quantification of various nodes is done by the user who is trying to protect his privacy. It represents his preferences in accordance with his view. However, they must obey the rules that we have listed. For example, users will want more privacy for locations at finer granularity than those at coarser granularity. Thus, the numbers for the nodes at the top of the location hierarchy will have values greater than nodes at the bottom.

5 Computing Disclosure and Privacy Levels

A naïve approach that works with all kinds of representation is to build a list of all possible contexts and associate a level of location disclosure with it. The context of the query posed by the user is matched with the set of contexts stored and the corresponding location information is returned to the user. The advantage of such an approach is that it is simple and gives the accurate location disclosure preference in the case of an exact match. Such a naive approach has several problems. First, we need to identify all possible contexts and associate location preferences for them. Second, the number of entries may be very large and it may not be efficient to search through them. Third, if the context of the query does not match any of the entries, the requester will receive default information. Once numerical values have been assigned to the different factors, we can have different techniques for calculating privacy preferences. One approach is to assign weights to each factor based on the preference. Let w_i, w_u, w_t and w_l be the weights assigned to the factors, namely, identity, usage, time, and location respectively, such that, $w_i + w_u + w_t + w_l = 1$ and $0 \leq w_i, w_u, w_t, w_l \leq 1$. The value of each factor is computed from the query context. Let v_i, v_u, v_t, and v_l be the values obtained for each factor specified in the query context. The level of information disclosure (L_p) is computed as follows.

$$L_p = w_i \times v_i + w_u \times v_u + w_t \times v_t + w_l \times v_l \tag{1}$$

The granularity at which location information can be disclosed is a function of the privacy preference. Higher values of L_p correspond to specifying location information at finer granularity. The level of information disclosure, L_p, is an input to the blurring function. The blurring function will be used to return the location information at the appropriate granularity level. The advantage of this approach is that it is not computationally expensive. The problem is that it requires the user to assign preferences to each of the factors. It is not always possible to form a total order among all the factors. For instance, for some requesters, the location of the user may have a higher importance than time of the day. For other requesters, it may be the opposite.

The next approach that we propose is a little different. Among all the factors that influence location privacy, role of the requester is perhaps the most important. We propose a scheme in which the policy owner considers three types of combinations: requester and usage, requester and time, and requester and location. For each of these combinations, he specifies his preference to disclose location information. In other words, we define three functions: $T_u : I \times U \rightarrow P$, $T_t : I \times T \rightarrow P$, and $T_l : I \times L \rightarrow P$ where $P \in [0, 1]$. The preference value 0 indicates the policy owner's unwillingness to disclose location information, and 1 indicates complete willingness to disclose location information. This allows each user to assign preferences to the combination of the requester and usage, requester and time, and requester and location. The importance of each combination is denoted by the weight factor. Let w_u, w_t and w_l be the three weights associated with the usage, time, and location factors corresponding to a given requester i. Here $0 \leq w_u, w_t, w_l \leq 1$ and $w_u + w_t + w_l = 1$. Let pu_{ij} be the preference associated with requester i and usage j, pt_{ik} be the preference associated with requester i and time k, and pl_{im} be the preference associated with requester i and location m. The level of information disclosure L_p is given by

$$L_p = w_u \times pu_{ij} + w_t \times pt_{ik} + w_l \times pl_{im} \tag{2}$$

Here again, we use the L_p as an input to the blurring function to return the location information at the correct granularity. We next show how L_p can be used with an existing

privacy preserving model, known as *k*-anonymity. Note that, although we demonstrate the use of our model in the in the context of *k*-anonymity, this does not preclude its use with other privacy preserving models and technologies. In the context of location-based privacy, *k*-anonymity requires that in response to a query about the location of an individual, the system provides a cloaked region where the location of the user is indistinguishable from *k* - 1 other users. The exact value of k depends on the personal privacy preference of the individual user, and also the context in which the query is issued.

In the following, we show how to derive the value of *k*, given the level of information disclosure L_p. Specifically, we provide a linear equation that relates *k* to L_p. For reasons of implementation, we impose an upper bound on the maximum value of *k* that is possible in any given system; we term this K_{max}. When the value of $L_p = 0$ (that is, user is not willing to disclose his location), the value of k equals K_{max}. When the value of $L_p = 1$ (the user is willing to disclose his exact location), the value of k equals 1. Using these two points, we obtain the following linear equation:

$$k = round\_to\_integer\{(1 - K_{max})L_p + K_{max}\} \tag{3}$$

Once we know the value of k, we show how it can be used for generating the cloaked region in response to a query. The implementation depends on the indexing technology used for managing the spatial database. Toward the generation of cloaked region, Gruteser and Grunwald have proposed an adaptive-interval cloaking algorithm that satisfies k-anonymity [11]. Their algorithm begins with the root node of the index tree and subdivides the area around the protected subject until the number of subjects in the area falls below k. Then the algorithm returns the parent quadrant as the cloaked region. We adopt this algorithm to generate the cloaked area that is used in our policy enforcement.

A graphic representation of adaptive-interval cloaking algorithm is shown in Figure 5. In this example, we assume that each of the nodes R13, R14, R11, R12, R8, R9, R10 contain a single user. When $k = 5$ and the protected user is located in node R11, the algorithm subdivides the area until it reaches quadrant R4 which has fewer subjects than 5. The algorithm then returns the cloaked region R1 which is the parent quadrant of R4.

6 System Architecture

Having presented our model for enforcing location privacy, we next describe the architecture of a system that ensures location privacy. We have tried to develop a system that can be used together with existing location-based services (LBS). We did not want our privacy module to interfere with the operations of existing services. Thus, one can think of our module as implementing location privacy as a service layer. This layer is implemented as a client server architecture model. The client module allows the users to manage their privacy policies through the *Access Management User Interface*. The server module consists of the *Privacy Agent* module that provides authorization services to the LBS. The client and the server module communicate through their own channel to avoid interference with LBS operations. We need to ensure that this channel is resistant to information leakage, impersonation, message spoofing and replay attacks. We will revisit these requirements and describe detail implementation of location privacy module in the Section 7. In this section, we simply describe the big picture of the system architecture including the functions and relations between sub modules. Our architecture is presented in Figure 6. It consists of three major components, namely, *Location Provider*, *Mobile Devices* and *Service Provider*.

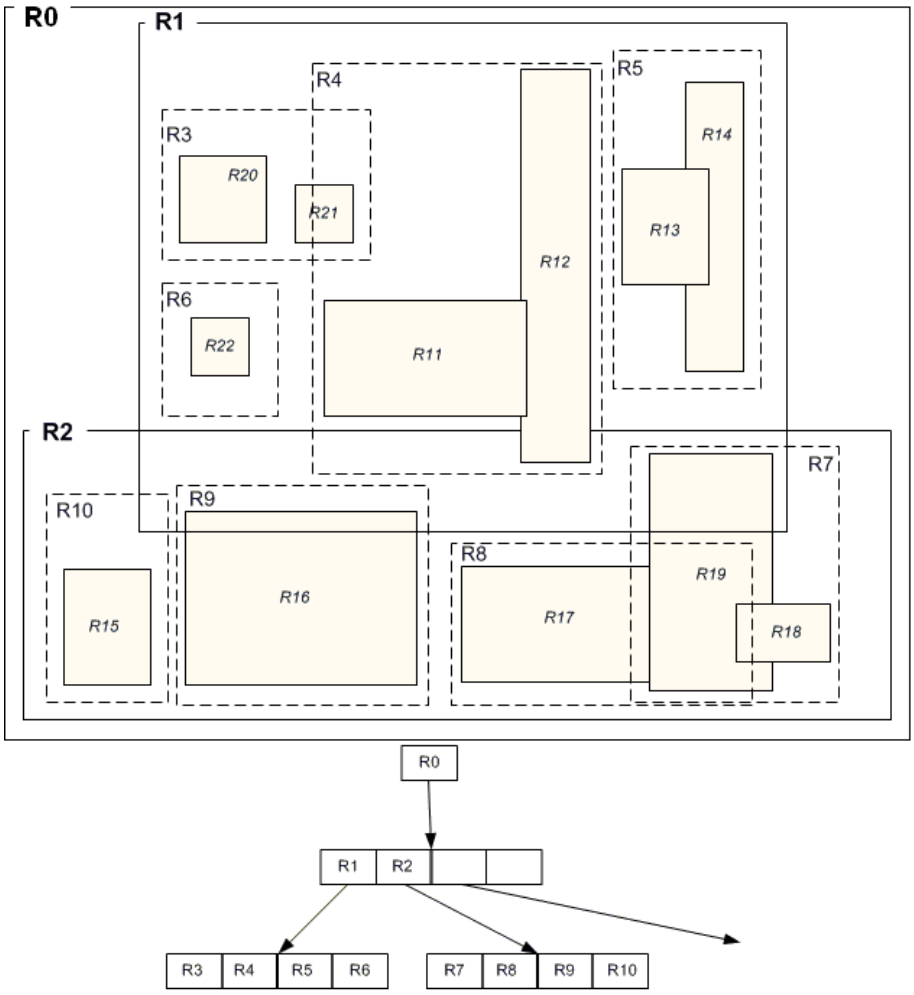

Figure 5: A graphical representation of an adaptive-interval cloaking algorithm

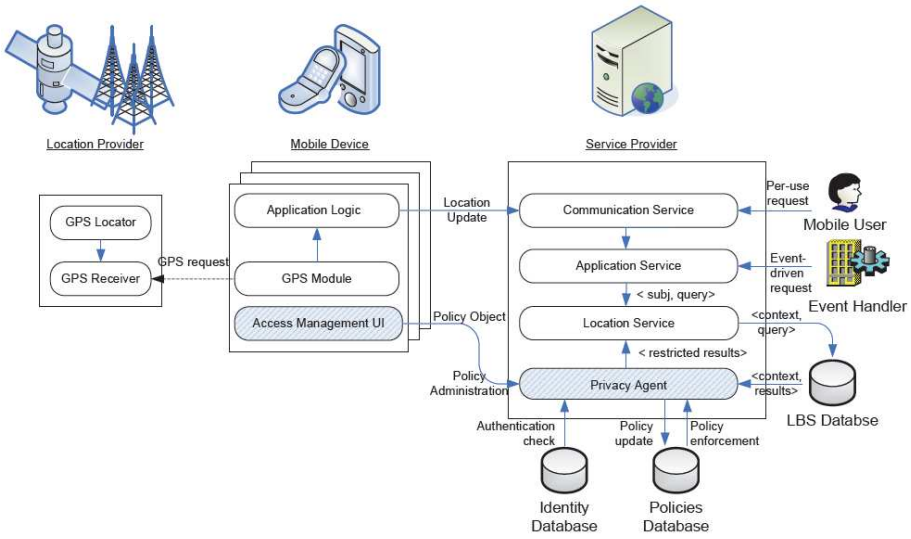

Figure 6: System Architecture

Location Provider: *Location Provider* computes the physical location of the user. It consists of the *GPS Receiver* which is responsible for receiving GPS requests from mobile devices and *GPS Locator* which computes the actual physical location. The results computed are passed back on to the querying mobile device.

Mobile Devices: Mobile devices include mobile phones, PDAs and other devices with GPS capabilities. Mobile devices use *GPS* module to find out its location. These location updates are sent to the trusted service providers, who are also responsible for providing necessary protections, such as those needed for location privacy, for the mobile devices. The *Application Logic* module manages the communication between the mobile device and the service providers. It sends location updates and receives event notifications from the service providers. In addition to these modules, our approach needs an *Access Management User Interface* module to allow mobile users to manage their own privacy. The *Access Management User Interface* allows the user to create and update location privacy policies. In addition, it allows the user to create, edit and manage context elements, such as user-role assignment. The *Access Management User Interface* directly communicates with the *Privacy Agent* module via the secure communication channel. The channel is used to send or update the policy object and/or context elements.

Service Provider: *Service Provider* provides location-based services to subscribed mobile devices. The services can be broadly classified into two groups based on the nature of the request: *event-driven requests* and *per-use requests*. Queries in event-driven service are issued internally in the *Application Service* module when a subscribed event occurs (e.g. two or more friends enter the same region) in the region where the user is currently located. On the contrary, per-use requests are often issued by mobile users for user-specific purpose (e.g. find if there is any gas station located in the range of 15 miles). External communications, such as location update, are handled by *Communication Service* module. The *Application Service* module provides different spatial services. It receives the requests either from the communication module or from its internal logic. It interprets the request,

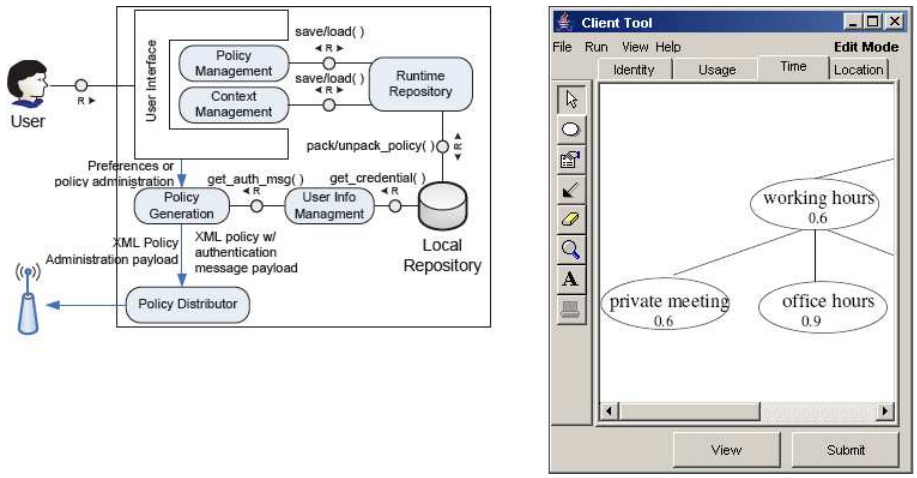

Figure 7: Access Management User Interface Architecture and Screenshot of Prototype Tool

generates an appropriate query, and passes the request to the *Location Service* module who processes it. In addition, we place a *Privacy Agent* module between location service and the *LBS Database*. The job of the *Privacy Agent* is to restrict the result such that it satisfies user's privacy. The *Privacy Agent* identifies the context of the query which is used for restricting the access to location information. The *Privacy Agent* enforces these restrictions by modifying the results (e.g. giving a cloaked region instead of the exact location) before responding to the query. The restricted results are sent back to the location service to execute the remaining part of the query before sending the privacy-aware results back to the requester (in the case of per-use request) or to the subscribers (in the case of event-driven service). The *Privacy Agent* is also responsible for updating policies. It receives the policy update request from mobile device, performs the authentication check, and updates the policy database accordingly.

7 Implementation

In this section, we describe the proof of concept implementation that we developed using Java. The implementation consists of two parts: a client application for managing the privacy policy (*Access Management User Interface*) and a server application that enforces the policy (*Privacy Agent*).

7.1 Access Management User Interface

We developed a J2EE application for mobile users to manage their privacy policy. The tool allows mobile users to create and manage their privacy policies. Figure 7 describes the tool that we have developed. The figure on the left describes the architecture of the *Access Management User Interface*, and that on the right shows a screen shot of the prototype tool. The major components in this architecture are described below.

Local Repository and Runtime Repository The policy objects are stored in the *Local Repository*. The *Local Repository* in our prototype tool is implemented using Microsoft Access

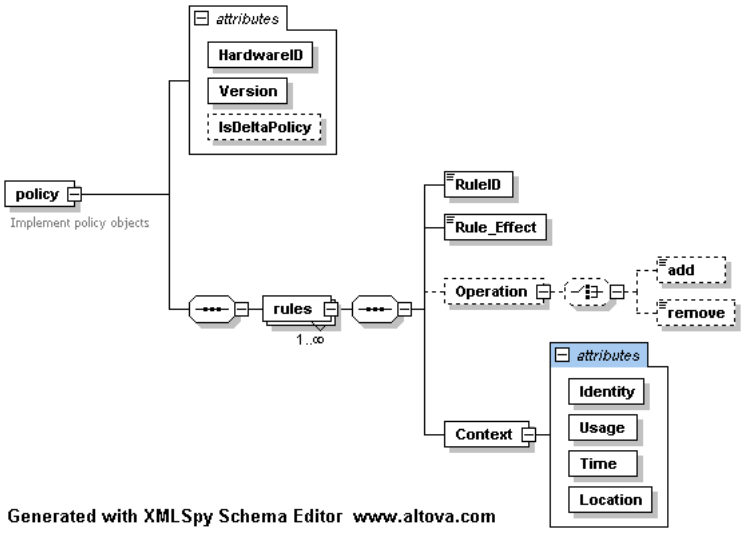

Figure 8: Policy Object Schema

database. The *Runtime Repository* interacts with the *Local Repository* and is responsible for caching the most frequently accessed policies. It is implemented using JDBC.

Policy Management This module is responsible for creating and editing policies. It sends requests to runtime repository service to get the policy objects from the local repository.

Context Management This module allows mobile users to create new elements in the contexts, edit them, and establish relationships among them. For example, it will allow the mobile users to assign new users to roles or change the hours designated as working hours.

User Interface The user interface around policy management and context management facilitates the user in creating and editing policies and contexts. Figure 7 shows a screen shot of the *Access Management User Interface* module.

User Info Management *User Info Management* module prepares the authentication message by encrypting the last 512 bytes of the policy with hardware ID. *User Info Management* also puts a nonce in the message to prevent replay attacks. This message is used for authenticating the request as well as for ensuring the integrity of the policy. This message is included in the same payload as the policy.

Policy Generation *Policy Generation* is responsible for transforming the privacy preferences of the user into a policy object and also computing the level of information disclosure L_p for every context. In addition, *Policy Generation* module is responsible for generating the XML message that contains the policy object together with the authentication message obtained from *User Info Manager*. The policies produced by the policy generator adheres to the schema shown in Figure 8.

Policy Distributor *Policy Distributor* is responsible for setting up the session with the server needed to send the policy object.

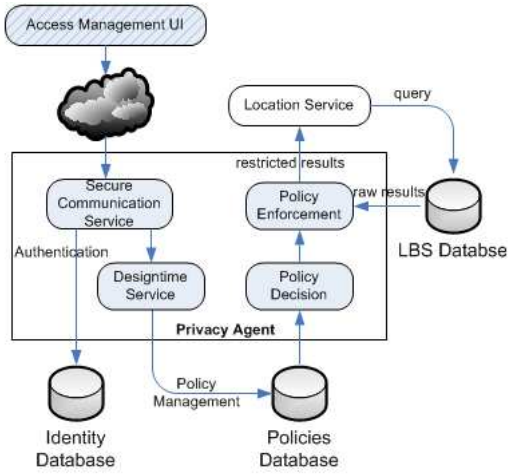

Figure 9: Architecture of the Privacy Agent Module

7.2 Privacy Agent

We also need to extend the server architecture to support our approach. The *Privacy Agent* module supports two modes of operation: design time and run time. The design time service describes the environment where the mobile users create or update their policies and modify context elements. The runtime service covers the policy decision and policy enforcement processes where *Privacy Agent* modifies the query results w.r.t. the user's privacy policy. Figure 9 shows the architecture of the privacy agent module. The major components in this architecture are described as follows.

Secure Communication Service *Secure Communication Service* is used at the design time. It receives requests for updating the policies or context elements, authenticates the requester's identity, and extracts the request from the message payloads. In order to validate the request, *Secure Communication Service* fetches the device's hardware ID from the identity database and uses it to decrypt the authentication message. Then it compares the decrypted message with the last 512 bytes of the message payload to ensure the integrity and authenticity of the message. It then sends the authorized requests to *Designtime Service*.

Designtime Service *Designtime Service* module receives authorized requests from *Secure Communication Service*. It then transforms the requests to policy management instructions (such as, add rules, delete rules, add user-role-assignment objects, and remove user-role-assignment objects) and updates the policies database.

Policy Enforcement *Policy Enforcement* module is used at runtime. It receives the query and the results produced by the LBS database, performs access restrictions using the help of *Policy Decision* module, and sends the modified results back to the location service. It identifies the context of the query and passes this information to the *Policy Decision* module. *Policy Decision* module responds with the level of information disclosure L_p. *Policy Enforcement* module then uses L_p with the underlying technology to return the appropriate query response. In our prototype, it uses L_p to obtain the desired value of k in the k-anonymity model, and uses cloaking algorithm to generate the response that satisfies this desired value of k.

Policy Decision *Policy Decision* module receives information about the context from the *Policy Enforcement* module, identifies the rule that best matches the context, and returns the level of information disclosure L_p to the *Policy Enforcement* module.

8 Simulation and Case Study

In this section, we first present some sample scenarios that describe how location-based queries will be answered in our system. Later we give some simulation results from our system.

8.1 Example Use Cases

In this section, we describe the basic use case scenarios that our prototype must support at design time or runtime. The use case scenarios at design time cover the basic operations of updating policies or context information, whereas those at runtime show how some basic queries are supported.

Use Case 1: Policy owner creates/updates the policy or the context information

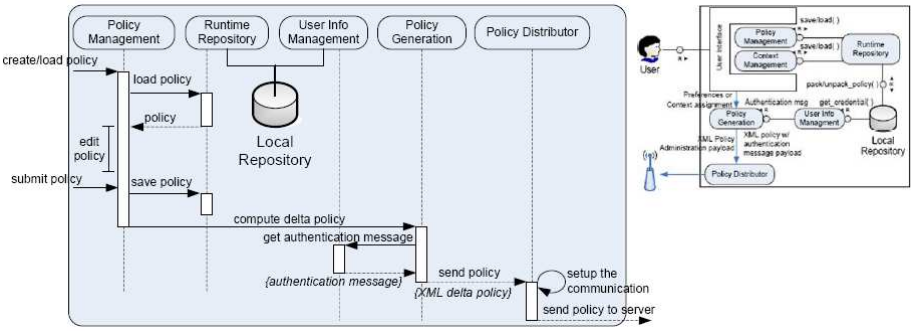

Figure 10: Sequence Diagram for Use Case 1

The mobile users create/update the policy or context information using the *Access Management User Interface*. If the users are updating existing objects, *Policy Management* module loads them from the local repository, edits them and submits them to the repository. It also sends the edited policy to *Policy Generation* module which converts it into XML format. *Policy Generation* module sends a request to *User Info Management* module to create the authentication message, which it then appends to the policy before forwarding to the policy distributor. *Policy Distributor* is responsible for sending the policy to the server.

Use Case 2: Querying the location of a specific user

In this case, *Policy Enforcement* mechanism queries the *LBS Database* to find the location of the user. *Policy Enforcement* mechanism identifies the context information and queries *Policy Decision* module about the level of information disclosure L_p. Then, L_p is used to compute the value of *k*, if the *k*-anonymity model is used, which then becomes input to the cloaking algorithm. The cloaked region is returned in this case.

Use Case 3: Finding the set of users in a given range

In this case, *Policy Enforcement* module queries the *LBS Database* to find all the users that satisfy the query. The context of the query is determined and the level of information

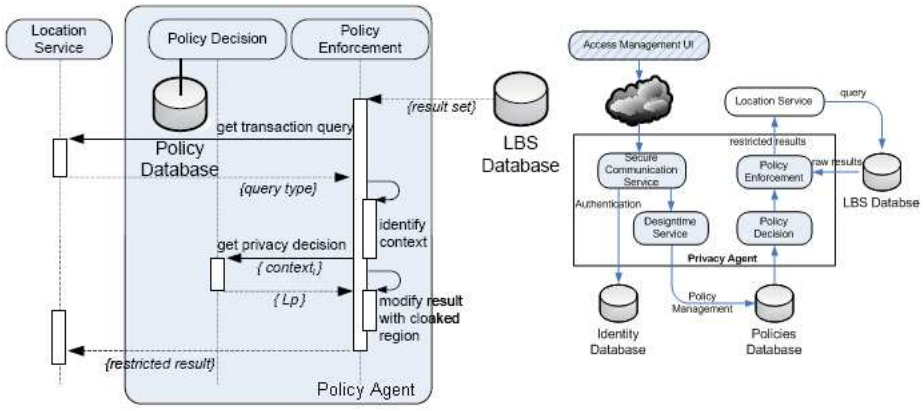

Figure 11: Sequence Diagram for Use Case 2

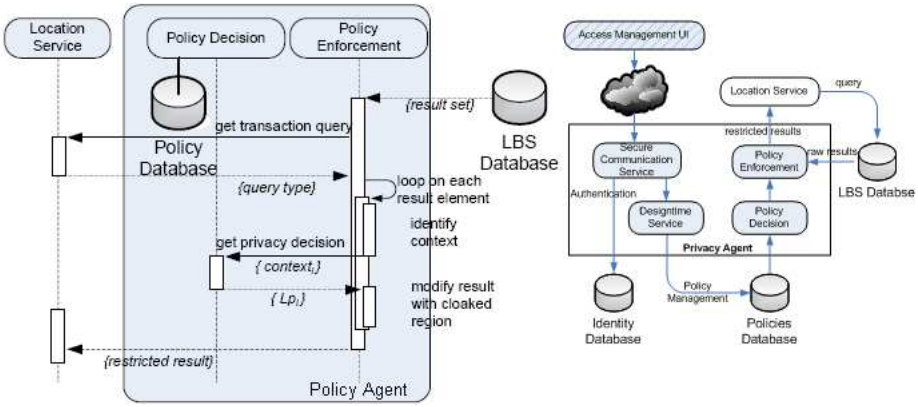

Figure 12: Sequence Diagram for Use Case 3

disclosure L_p permitted for each user is computed. Using L_p, the value of k is computed for each user that satisfies the query. If the value of k for some user *UserP*, exceeds the number of users that will be reported back, then *UserP* is removed from the result set. Otherwise *UserP* is included in the result set.

Use Case 4: Find p users that are in the vicinity of the requester This case is handled as follows. We find the smallest node containing the requester's location and see if it contains p users. If not, we look at the parent node, to find p users, and repeat this process until p users are found. For each of the p users, we compute the value of k using the context of the query and the privacy preference of the individual user. For any user, if the value of k is less than p, then this user is included in the set. Otherwise, it is not included. Finally, we count the total number of users who have a value of k less than p. If this set equals p, then it is revealed to the requester. If this set is less than p, then we go up one level to find the parent node, to include more children.

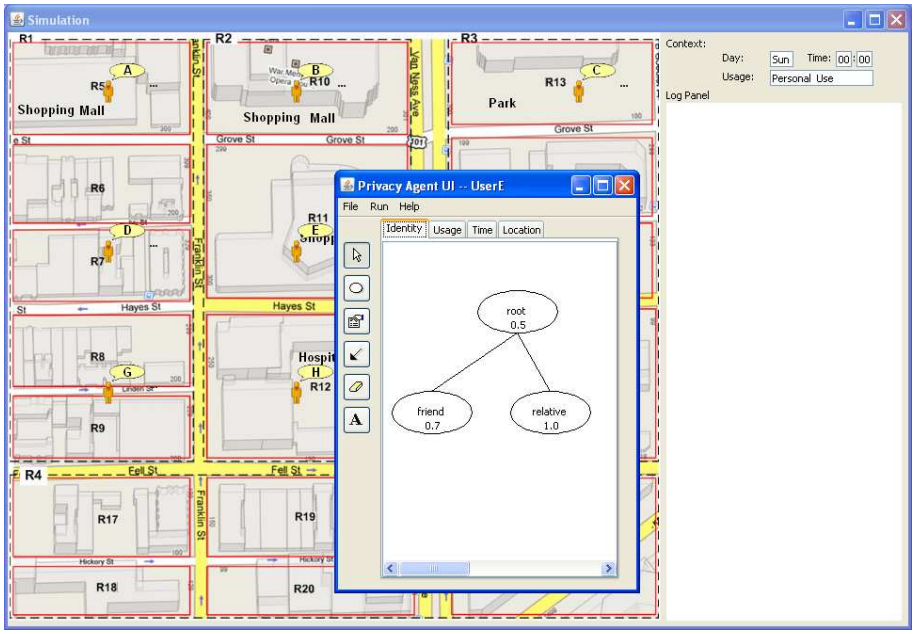

Figure 13: Access Management in Simulation Program

8.2 Simulation

We developed a JAVA simulation program to illustrate the workings of our *Access Management User Interface* and *Privacy Agent* modules. The program assumes the existence of some fictitious spatial services covering the San Francisco area (obtained from Google Map). We have 10 users, denoted by *UserA*, *UserB* etc. Users can be moved to different locations via mouse drag and drop. The program also allows each user to create his privacy preferences using *Access Management User Interface* (see Figure 13). The policy decision process and the simple cloaking algorithm based on k-anonymity imitates the process of privacy-aware LBS that exists in the *Privacy Agent*.

We tested our program using two scenarios. In the first scenario, *UserE* specifies his privacy policy as shown in Figure 14. He assigns equal weight to identity, usage, location, and time factors. He divides the identity space into *Friend* and *Relative* and assigns them values of 0.7 and 1.0 respectively. The root has a value of 0.5. If the requester is neither friend nor relative, he gets the access privilege of this root node. In the context of usage, he divides the space up into *Personal_Use*, *Business*, and *Emergency*. Here again, he assigns values 0.7, 0.6, and 1.0 respectively. The other usages are assigned to the root node which has a value of 0.5 by default. He organizes his time into *Business*, *Private* and *Vacation* and the values assigned to them are 1.0, 0.25, and 0.6. He also categorizes his location information into *Hospital*, *Office*, *Shopping_Mall* and assigns values 0.25, 0.9, and 0.5 respectively.

UserE assigns *UserD* and *UserC* as his relative and denotes *UserA*, *UserB* and *UserK* as his friend. In this simulation, he sets up the period 00:00 to 08:59 on Monday as *Private*, 9:00 to 16:59 on Monday as Business, and 17:00 to 23:59 on Monday as *Vacation*. For simplicity, the other times are not specified and they would be assigned the root node. He assigns building R12 to *Hospital*, R5, R10, R11, R16, R21 to *Shopping Mall*, and the other buildings

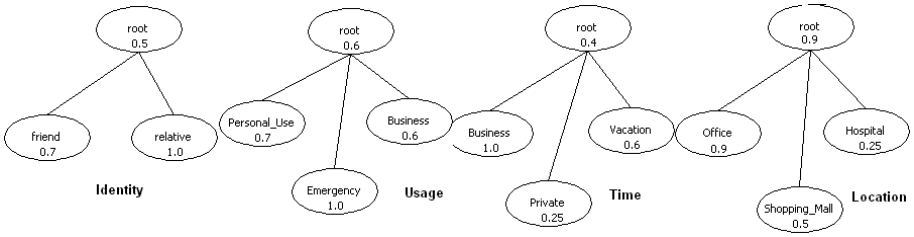

Figure 14: *UserE*'s Privacy Policy

not labeled in Figure 15 as *Office*.

UserD issues the following query on Monday 13:00 for personal use: find all users that are within 200 ft. distance from him. The LBS finds *UserA*, *UserB*, *UserE*, and *UserG* in the range of the query. The LBS then finds whether any of these users have privacy policies set up with the privacy agent or not. In this case, only *UserE* has a privacy policy, so we need to identify the context in which the query was issued. It identifies the context as $< Relative, Personal\_Use, Business, Shopping\_Mall >$. Since equal weight has been assigned to all the factors, the value of L_p and k are computed as follows:
$L_p = 0.25 \times 1.0 + 0.25 \times 0.7 + 0.25 \times 1.0 + 0.25 \times 0.5 = 0.8$
$k = (1 - 7) \times 0.8 + 7 = 2;\ assuming\ K_{max} = 7$
The final step is to ensure that *UserD*, the requester, cannot distinguish *UserE* from another user. In this case, we see that the *k*–anonymity requirement is satisfied and *UserE* together with the other users are returned in the result set.

In the next scenario, the same query is issued by *UserJ* an hour later while *UserE* is in the hospital (R12). The privacy agent identifies the context instance as <*root*, *Personal_Use*, *Business*, *Hospital*>. The level of privacy disclosure L_p computed by Equation 1 is 0.61 and k is increased to 3 (see Figure 16). Hence, the search range no longer satisfies the *k*-anonymity criteria. So the *Privacy Agent* removes *UserE* from the result set.

9 Security Analysis

We define a location privacy threat as an incident in which an adversary can identify a one-to-one mapping between an individual and location information. In our case, attackers may be legitimate users of the system or outsiders. In general, the following attacks may occur in our system (please refer to the system architecture given in Figure 6): *system penetration*, *denial of service*, *insider attacks*, *service spoofing*, *eavesdropping*, *message altering*, *replay attacks*, *masquerading*, and *location tracking*.

We assume that the service provider is secure and trusted. We also assume that the communications with the service provider have adequate security protection. Consequently, we do not consider attacks such as system penetration attacks, insider attacks, denial of service attacks, service spoofing, eavesdropping or message altering. We do not elaborate any further on these attacks, but describe the other attacks that are possible in our system.

Revealing User's Preferences

User's privacy preferences are stored in mobile device. The preference is distributed to the service provider through policy update communication. Attacker can capture these messages and may want to discover user's preferences. For this to constitute a privacy

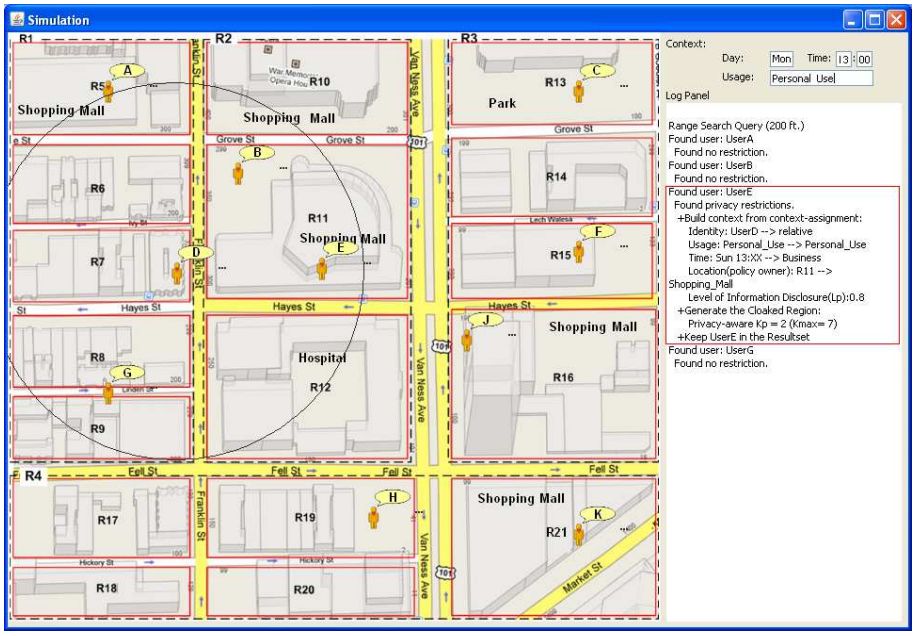

Figure 15: Range Query Issued by *UserD* querying for persons who are within 200 ft.

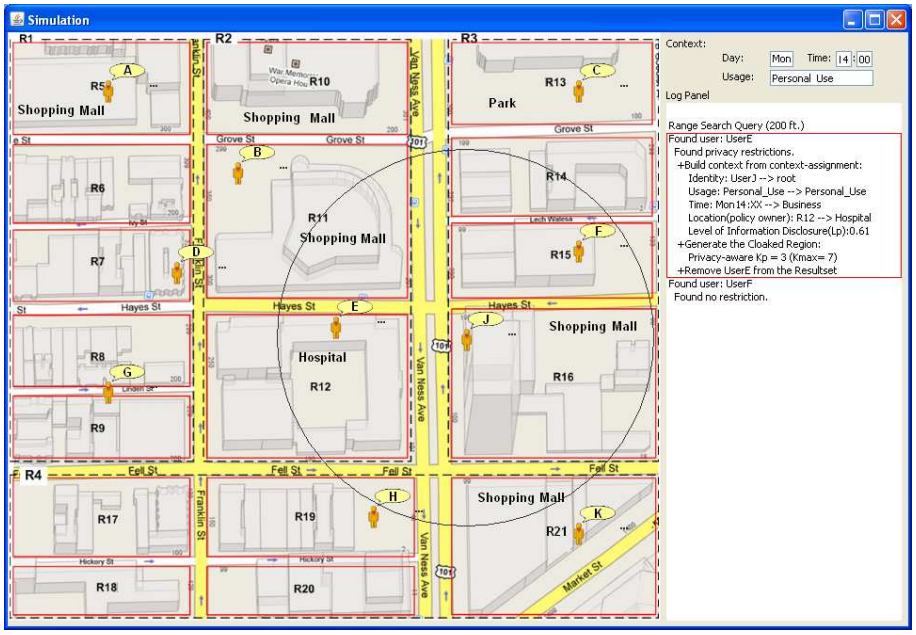

Figure 16: Same Query Issued by *UserJ* while *UserE* is in the Hospital

threat, the adversary must be able to correctly identify the hierarchical structure of user privacy preferences as well as identity of the mobile user. Our approach is resistant to this type of attack. This is because the L_p is computed locally and the mobile user only distributes projections of his privacy preference hierarchies. The projection is in the form of a 5-tuple $< I, U, T, L, L_p >$. It is not possible to reconstruct the original hierarchy from the set of tuples since neither parent-child nor dominance relationship can be derived from the set of tuples alone. In fact, even the attacker can compare L_p between the two tuples which only have one attribute different (either I, U, T, or L), the ΔL_p is still a product of two variables; weight (w) and preference value (v). These variables are known only to the policy owner. Hence, an attacker can not correctly identify the user's preferences.

Policy Message Altering

Adversary can intercept and alter the policy update message. This is a specific case of the message altering attack which has a race condition. If an attacker can alter the privacy policy of a user, then location privacy becomes severely compromised. Thus, we need to completely eliminate this threat. To eliminate this threat, we introduce the authentication message. The authentication message is generated by encrypting the last 512 bytes of the policy with user's hardware ID. The encrypted message is decrypted by Service Provider to verify message integrity and authenticity. Thus, attackers cannot alter the policy as long as the key is not compromised.

Replay Attack

Attacker can capture the communication message between legitimate user and service provider and later resend the message to the server. Our model prevents replay attack by including a nonce in the authentication message. Attacker may capture the policy update message but can not resend the policy update to alter the original policy stored in the server.

Masquerade

Attacker can impersonate as a legitimate user and inject spoofed policy to the server. We eliminate this threat by attaching an authentication message to the policy update. The authentication message is used for authenticating the request as well as for ensuring the integrity of the policy.

Location Tracking

An adversary may obtain the current position of an individual through the LBS. Continuous tracking this information allows him to track the movement of individual. The proposed model employs Gruteser's k-anonymity cloaking algorithm to generate the cloaked region. The cloaked region prevents an adversary from correctly identify a one-to-one mapping between user's identity, timestamp and location. However, the proposed model does not completely eliminate the threat. This is because the model does not hide the direction and we have no control over other public information which can be used to infer sensitive information. For example, in Palo Alto, Veteran Hospital is surrounded by IT research facilities. If the cloaked region covers the location of the hospital and adversary knows that the target does not work in a high technology business, then location privacy may be compromised.

10 Conclusion

Technological advancements in mobile computing have spawned a growth in location-based services. Such services use the location information of the subscriber to provide better functionalities. Improper usage of location information may compromise the secu-

rity and privacy of an individual. Moreover, a user must be allowed to control who has access to his location information and under what circumstances. Towards this end, we investigate the factors influencing location privacy, suggest techniques for quantifying them, and propose different approaches for expressing the user's privacy preference with respect to the disclosure of location information. The approaches differ with respect to the storage requirements, and the granularity of privacy preference. We also presented the architecture of the system that allows for location privacy. We have developed a prototype that illustrates the feasibility of our ideas.

A lot of work remains to be done. In this paper, we have not provided any guidelines to the user for specifying their privacy preferences. We plan to do some empirical studies using real world users and see how they specify their preferences and quantify the various factors that will affect the disclosure of their location information. We also need to validate our model and enforcement mechanism for preserving location privacy using real world data.

In this paper, we have shown that how the level of information disclosure can be mapped to an existing privacy model, such as *k*-anonymity model. In future, we need to illustrate how it can be used in the context of other privacy preserving technologies. We also need to develop more detailed implementation pertaining to our model. Specifically, we need to understand how location information is stored and managed in our model. Next, we need to validate our model using real-world applications and data. We plan to do this as part of our future work.

Acknowledgments

The work was supported in part by AFOSR under contract number FA9550-07-1-0042.

References

[1] E. Bertino, B. Catania, M.L. Damiani, and P. Perlasca. GEO-RBAC: a spatially aware RBAC. In *Proceedings of the 10th ACM Symposium on Access Control Models and Technologies*, pages 29–37, 2005.

[2] Suroop Mohan Chandran and James B. D. Joshi. *LoT-RBAC*: A Location and Time-Based RBAC Model. In *Proceedings of the 6th International Conference on Web Information Systems Engineering*, pages 361–375, New York, NY, USA, November 2005.

[3] C.Y. Chow, M.F. Mokbel, and X. Liu. A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Service. *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*, pages 171–178, 2006.

[4] S. Consolvo, I. E. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge. Location disclosure to social relations: why, when, & what people want to share. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 81–90, New York, NY, USA, April 2005. ACM Press.

[5] L.F. Cranor. *Web Privacy with P3P*. O'Reilly Media, Inc., 2002.

[6] L.F. Cranor. P3P: Making Privacy Policies More Useful. *IEEE Security & Privacy*, pages 50–55, 2003.

[7] J. Cuellar, J. Morris, and D. Mulligan. RFC 4079: Geopriv requirements. *Internet Engineering Task Force (IETF) Internet Draft. http://www. ietf. org/ids. by. wg/geopriv. html*, 2003.

[8] M.L. Damiani, E. Bertino, B. Catania, and P. Perlasca. GEO-RBAC: A spatially aware RBAC. *ACM Transactions on Information and System Security*, 10(1), 2007.

[9] B. Gedik and L. Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. *Proceedings of the 25th International Conference on Distributed Computing Systems*, pages 620–629, 2005.

[10] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.L. Tan. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 121–132, New York, NY, USA, 2008. ACM.

[11] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pages 31–42, 2003.

[12] Urs Hengartner and Peter Steenkiste. Implementing Access Control to People Location Information. In *Proceeding of the 9th Symposium on Access Control Models and Technologies*, Yorktown Heights, New York, June 2004.

[13] T. Jiang, H.J. Wang, and Y.C. Hu. Preserving location privacy in wireless lans. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, pages 246–257, 2007.

[14] H. Kido, Y. Yanagisawa, and T. Satoh. An Anonymous Communication Technique using Dummies for Location-based Services. *Proceedings of IEEE International Conference on Pervasive Services*, pages 88–97, 2005.

[15] Ulf Leonhardt and Jeff Magee. Security Consideration for a Distributed Location Service. *Imperial College of Science, Technology and Medicine, London, UK*, 1997.

[16] M.F. Mokbel, C.Y. Chow, and W.G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 763–774, September 2006.

[17] L. Palen and P. Dourish. Unpacking" privacy" for a networked world. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 129–136, 2003.

[18] N. Poolsappasit and I. Ray. Towards a scalable model for location privacy. In *Proceedings of the 1st ACM GIS Workshp on Security and Privacy in GIS and LBS*. ACM Press, November 2008.

[19] I. Ray and M. Toahchoodee. A Spatio-Temporal Role-Based Access Control Model. In *Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, pages 211–226, Redondo Beach, CA, July 2007.

[20] Indrakshi Ray, Mahendra Kumar, and Lijun Yu. LRBAC: A Location-Aware Role-Based Access Control Model. In *Proceedings of the 2nd International Conference on Information Systems Security*, pages 147–161, Kolkata, India, December 2006.

[21] Indrakshi Ray and Manachai Toahchoodee. A Spatio-Temporal Access Control Model Supporting Delegation for Pervasive Computing Applications. In *Proceedings of the 5th International Conference on Trust, Privacy & Security in Digital Business (to appear)*, Turin, Italy, September 2008.

[22] K. Ren, W. Lou, K. Kim, and R. Deng. A Novel Privacy Preserving Authentication and Access Control Scheme for Pervasive Computing Environments. *IEEE Transactions on Vehicular Technology*, 55(4), 2006.

[23] I. Smith, S. Consolvo, J. Hightower, J. Hughes, G. Iachello, A. LaMarca, J. Scott, T. Sohn, and G. Abowd. Social Disclosure of Place: From Location Technology to Communication Practice. *Proceedings of the 3rd International Pervasive Computing Conference*, pages 134–151, 2005.

[24] E. Snekkenes. Concepts for personal location privacy policies. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 48–57, New York, NY, USA, 2001. ACM Press.

[25] W.H. Stufflebeam, A.I. Antón, Q. He, and N. Jain. Specifying privacy policies with P3P and EPAL: lessons learned. In *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, pages 35–35, 2004.