Summer 2011

Location privacy policy management system

Arej Awodha Muhammed

Department:

LOCATION PRIVACY POLICY MANAGEMENT SYSTEM

by

AREJ AWODHA MUHAMMED

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER SCIENCE

2011

Approved by

Dr. Dan Lin, Advisor
Dr. Srirarm Chellappan
Dr. Sahra Sedigh

ABSTRACT

The advance in wireless communication and positioning systems has permitted development of a large variety of location-based services that, for example, can help people easily locate family members or find nearest gas station or restaurant. As location-based services become more and more popular, concerns are growing about the misuse of location information by malicious parties. In order to preserve location privacy, many efforts have been devoted to preventing service providers from determining users' exact locations. Few works have sought to help users manage their privacy preferences; however management of privacy is an important issue in real applications. This work developed an easy-to-use location privacy management system. Specifically, it defines a succinct yet expressive location privacy policy constructs that can be easily understood by ordinary users. The system provides various policy management functions including policy composition, policy conflict detection, and policy recommendation. Policy composition allows users to insert and delete policies. Policy conflict detection will automatically check conflict among policies whenever there is any change. The policy recommendation system will generate recommended policies based on users' basic requirements in order to reduce users' burden. A system prototype has been implemented and evaluated in terms of both efficiency and effectiveness.

ACKNOWLEDGEMENTS

First, I thank Allah, who blessed me with courage and devotion for this study, and then I would like to express gratitude to my family especially my parents for their constant support and prayers during the time of my study. I owe acknowledgment, to my brother, Farag who has given me the confidence and without his constructive suggestions, and invaluable guidance, this study would not be feasible.

Also, I'm very grateful to my adviser, Dan Lin, whose continued guidance, encouragement, and support has shaped this thesis. I would like to thank my fellow students and faculty in my department for their helpful comments and feedback. In particular, I would like to express my appreciation to our department chair, Dr. Ali Hurson, and my committee, Dr. Sahra Sedigh, and Srirarm Chellappan, I thank them for their valuable time and efforts for this study.

Lastly, I offer my regards to all of those who supported me in any respect during the completion of study.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Page

1. INTRODUCTION

Advances in wireless communication and positioning systems (e.g., GPS) have permitted development of a large variety of location-based services (LBS). Such services may tell users the waiting time for a table in a nearby restaurant, or tell user when friends are located within walking distance.

As location-based services become more and more popular, concerns are increasing about misuse of location information by malicious parties. Since LBS providers can now continuously track and transmit a user's location information, users receive LBSs at the expense of sacrificing their location privacy. Location-detection devices pose a major privacy threat on its users where it tracks and transmits private information. Exposure of location information may put users into the danger of criminal behaviors. Kidnappers could take advantage of LBSs to acquire a target's location information. Also, it is possible that your moving device such as cell phones will be chocked by junk message from advertisers if there is a bonding of phone numbers and location services. Some users will not be willing to disclose location information in particular circumstances in particular time. Location privacy has been a major concern due to its common use in our daily life. For example, a user may want to know the waiting time for a table in a nearby restaurant. User may also wish to be notified when friends are located within walking distance. All these applications require an extensive use of location data [1]. Many governments and organizations have initiated studies of location privacy. For instance, in 1890 US Supreme Court Justice Louis Brandeis stated

that “the right to be left alone” is one of the fundamental rights of a democracy. US government has recently initiated a discussion of privacy in connection with the Location Privacy Protection Act [2]. The Internet Engineering Task Force (IETF) Geoprive working group [3] is also studying the requirements of location privacy. In summary, location privacy refers to the right of individuals to decide how, when, and for what purposes their location information can be disclosed to other parties. The lack of location privacy protection can be misused by a malicious party to launch attacks without the user’s consent, compromising the user’s personal privacy and security. Serious privacy issues must be addressed to satisfy both public concern and the need for compliance with current legislation.

To decrease the efforts to track an individual’s movements, many location privacy protection methods have been proposed. Information privacy concerns have mounted globally [4,5]. Issues range from detailed, publicly available satellite imagery over collection on the internet to DNA database. Most of existing efforts focus on preventing service providers from knowing users’ exact locations. Privacy could be protected in such applications by rendering the data anonymous before sharing it with application service providers. An anonymous location dataset provides strong privacy protection while allowing sharing with arbitrary data consumers, since no purpose-binding restricts the data for certain uses. However, this approach requires techniques beyond omitting obvious identifiers, since the spatiotemporal characteristics of the data allow tracking and re-identification of anonymous vehicles when user density is low. Therefore, a common strategy is to use an anonymization agent [6, 7, 8] to distort users’

real locations and send the distorted locations to the service provider. The service provider will directly process the distorted data to answer location-based queries. The anonymization agent is responsible for converting the query results back to the form that the end-users can understand. For example, in [9, 10], the anonymization agent will "cloak" users' locations before sending them to the LBS by providing their location at a lower resolution in terms of time and space. In other words, rather than giving a precise location and time instant, the agent would report a larger region covered in a time frame. K-anonymity has also been used often to measure general privacy [11,12-22]. It requires that each user can report his location only when there are more than k-1 other users in the same region. These approaches help improve location privacy of end-users.

Despite extensive works on protecting users privacy against service providers, few works have been done to help users manage their privacy preferences; however management of privacy is an important issue to be addressed in real applications. In [23,24-29], some simple location privacy policies were suggested to govern who can use an individual's location data under what conditions. However, to the best of our knowledge, there is not a comprehensive system for location privacy policy management. This work developed an easy-to-use location privacy management system. Specifically, it defines a succinct yet expressive location privacy policy constructs that can be easily understood by ordinary users. The system provides various policy management functions including policy composition, policy conflict detection, and policy recommendation. Policy composition allows users to insert and delete policies. Policy conflict detection will automatically check conflict among policies whenever there is any change. Policy recommendation system will generate recommended policies based on users' basic

requirements in order to reduce users' burden. A system prototype has been implemented and evaluated in terms of both efficiency and effectiveness.

Figure 1.1. Illustrates an overview of the proposed location privacy policy management (LPPM) system. The system is a policy-based location privacy system. All aspect of the system are comprised of small packets of rules, called policies. Its main job is to display the location of users to one another. The system is comprised of policies, with each type of policy producing its own information, and how that information is shared. The main point of the system is to take policies in and using the information that the differing type of policy provide, create a report that distinguish what information that each of the relation that a person has within the system can see. Policies are set up so that their information is reported at specific times of the day, and on specific days. Policies in the system can be added, modified and deleted at any time, allowing users a large amount of customization. As policies are entered into the system, they are compared to established policies; to ensure that they do not conflict. Policies are also compared to established policies to determine whether they can be merged together. The user interface is set up to be convenient and clear for the user. By evaluating a prototype of the system, a baseline of performance for a fully functional system can be established. In other word, the prototypes worth can be used to estimate the worth of a working system. The evaluation of the prototype of the location privacy system will show the capability of the system and with that one can see the usefulness of the location privacy system.

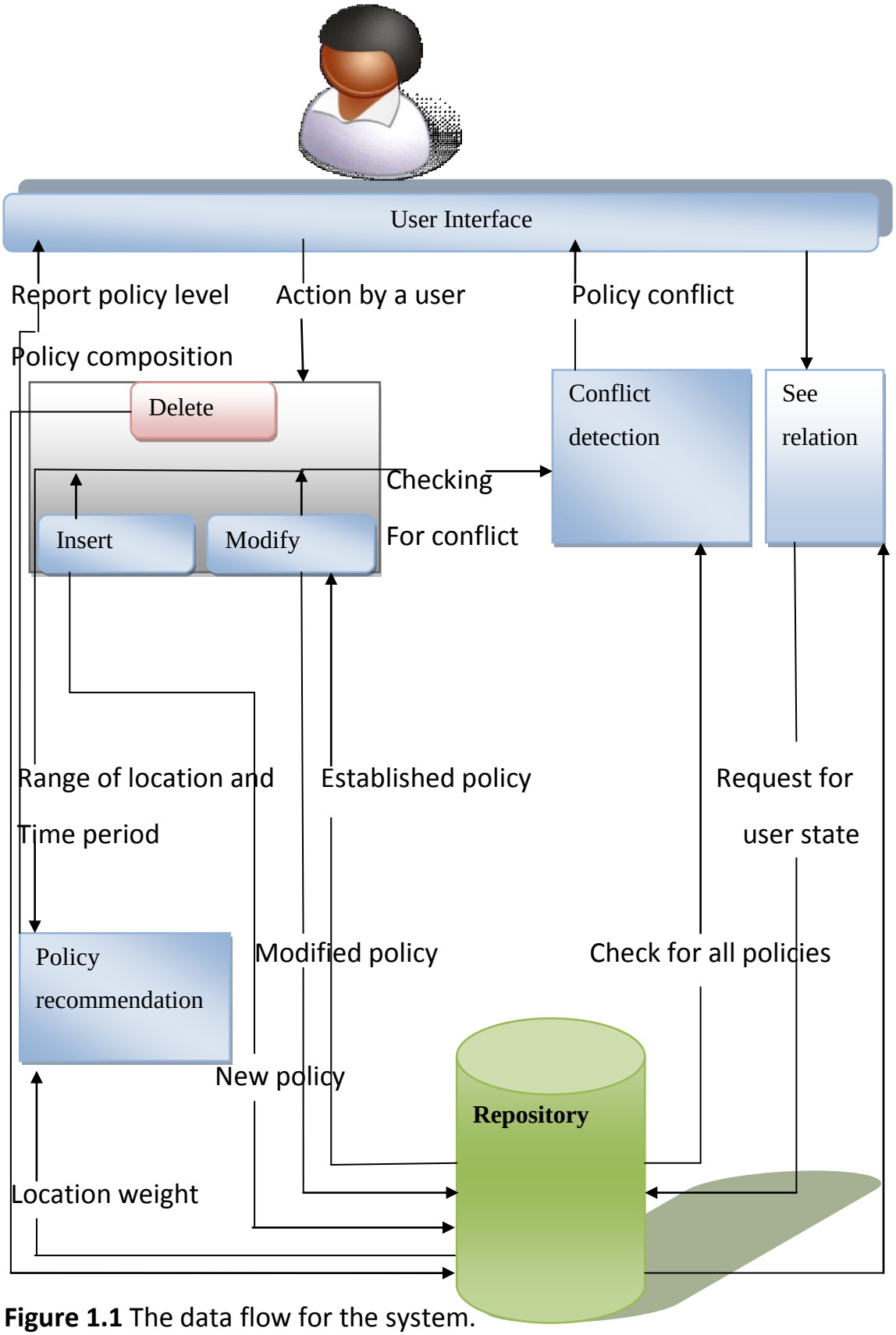

Figure 1.1 The data flow for the system.

2. LITERATURE REVIEW

Location information has been defined as a set of data associated with an individual that describe their location over a period of time [49]. The time resolution and location resolution vary with the technology used to collect them. There are several methods which have been used to locate the user. One the earliest system designed for location tracking is Global Positioning System (GPS) and WLAN (Wireless Local Area Network) [50].

Most of the time people are not concerned about their location privacy. However, they are sensitive to how their location data could be used, and this sensitivity could be increased with their awareness of privacy leaks [50]. Kaasinen made interviews with 55 people divided into 13 groups from different parts and different backgrounds . The result of this evaluation is that users trust current service providers and policy-makers for issues related to privacy protection.

J. Krumm in his project; he convinced over 250 people from his institution to give them two weeks of GPS data recorded in their car. He asked 97 of them whether is it possible to share their location data outside their institution, and the result was only 20% of them said “no” [51].

Danezis et al. [49] asked 74 undergraduates how much they would have to be paid to share a month’s worth of their location data. The average price was 10£, or 20£. The major problem in location privacy may occur during the communication between the user and the service provider; while a user is providing information to the SB the attacker may take advantage of this location information. Andreas et al [52]. Classify the attack into two types first-hand communication where an attacker obtains private information

through first-hand communication when an individual unwittingly provides it directly to the attacker. Second-hand communication:- attackers relay information from one party to another unauthorized party; in this one an individual no longer controls the information.

Many efforts have been made to ensure users' location privacy when they enjoying LBSs. The approaches that researchers have proposed can be categorized as policy-based, encryption-based, obfuscation-based, and anonymity-based approach.

2.1. POLICY-BASED APPROACH

Privacy policies are legal notices that contain statements defining what service providers can do with a user's personal data. Privacy policies are published by service providers, and users decide whether such policies are acceptable to them. These policies address many concepts and specific language is used to define them [23, 30]. Users reach an agreement with providers about which data are collected, what these data are used for and how they can be distributed to third parties. In this technique, privacy is understood as the ability of individuals to decide when, what, and how information about them is disclosed to others. Ideally, users can choose among various policies. Therefore, depending on the selected policy, users can save some money but, on the other hand, providers can distribute/sell some of their data.

This approach is easy to use because it not only satisfies user's personalized privacy requirements; but also frees the user from the continuous intervention by the system would otherwise require to address specific cases.

Smailagic et al.[31] proposed a privacy model and methodology based on the context-aware system Portable Help Desk. The privacy of location information is

described by set theory and rules. Each rule establishes a list of users who are allowed or disallowed to know the location of a user for a given duration of time. A rule establishes one time duration and possible repetition of an event. The rule sets authorization based on one of four visibilities, *Visible to All, Invisible to Some, Visible to Some, and Invisible to All*. These visibilities are arranged as increasing restrictiveness of the set. Visible to All allows anyone to know the location of a client user, Invisible to Some restricts only a finite list of users, Visible to Some restricts all users except a finite number of users, and Invisible to All restricts everyone. The conclusion shows a distribution where twice as many people appear willing to automatically transmit their personal information to any user, and a second group chooses greater privacy by rigorous setup of who is allowed to inquire of their personal information. Few users choose to grant or deny access to information separately for every request. To resolve conflicts between multiple privacy rules; if one rule states that user A is authorized to see a client's location at a given time and another rule states the opposite, which rule will take precedence. The researchers described such a situation to the users; they could respond by including "Cannot See" meaning the client wishes their information not to be given if a conflict exists, and "Can See" meaning the client wishes their information to be given if such a conflict exists.

Myles et al. [32] created a unifying location service called Loc-Serv, a middleware service that lies between location-based applications and location tracking technologies. The primary purpose of LocServ is to unify location tracking technologies so that a location-based application can make use of multiple positioning systems. In

essence, users of LocServ can specify a location query using any of the symbolic or geometric location models that Loc-Serv understands. The system requires a mechanism for controlling access to users' location information without the needs for repeated user intervention. It employs the same basic concepts used in P3P and pawS, using machine-readable privacy policies and user preferences to automate the process of deciding whether or not a particular piece of location information can be released to a third-party. Their approach can reduce the load on the user associated with processing requests for their location information. Users can enjoy location services without continuous interruptions from the system requesting permission to disclose current location information. Also, the authors proposed to define private policies under ubiquitous scenarios instead of just establishing simple privacy models for specific fields such as business or traffic, as previous researchers had proposed. However, they do not consider that collecting user's decision for various scenarios is itself a heavy load for the user, especially in some circumstances when location privacy may not be a great concern to the user.

Snekkeness [23] proposed that individuals should be equipped with tools that would allow them to formulate their own personal location privacy policies, subject to applicable rules and regulations. The author identifies concepts that may be useful when formulating such policies. The key concept is related to observation of a located object. An observation typically includes the location, the identity of the object, the time the observation was made, and the speed of the object. The idea is that the individual should be able to adjust the accuracy with which these observations are released, depending on

parameters such as the intended use of the information and the identity of the recipient. This approach provides fragments of a language for formulating personal location privacy policies. The main idea is that there should be established some register at some well-known address, that for each located object contains a pointer to the location where a personal location privacy policy for that located object is stored. A location provider would then be obliged to receive 'release approval' from the policy custodian before any location data could be released. Since location privacy is based on some well-known address, the user's location privacy may not be protected when the user is in a private location such as residence. The drawback of this approach is that it does not provide users with full protection, despite the variance of location.

Related research in role-based access control (RBAC) has recently received considerable attention, and it can also be used to maintain location privacy. In RBAC, permissions are associated with roles; users are assigned appropriate roles, and they acquire the granted to such roles' permissions. This approach greatly simplifies the management of permissions. Roles can be created for the various functions of an organization, and users can then be assigned to roles based on their responsibilities and qualifications. Users can be easily reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, and permissions can be revoked for particular roles as needed. In location privacy we can define roles such as supervisor, family, friends and strangers. Each role is assigned with different permission to user's location information depending on the request time and

location. In this way, individual's personal location privacy can be maintained and also we don't have to worry about the large numbers of various policies. However, since it role-based access control, a limited number and meaning of roles may not satisfy various needs of users for various situations in everyday life.

The goal of this approach is to satisfy a user's personal privacy requirements. In order to achieve this goal, a large number of policies are indispensible. As the number of users increases, the number of policies may also increase in an exponential manner. This makes policy-based approach not as scalable as encryption-based approaches.

The overhead of this approach comes from maintaining these policies. A large storage space is needed to keep record of individual user's privacy policy, and each time a user acknowledges a request, the server has to check with the policy and then an action can be taken according to individual's policy. This requirement creates enormous overhead, especially when users have detailed and complex policies.

2.2. ANONYMITY APPROACH

Anonymity-based approaches are the most sophisticated option in trusted-third-party-based location privacy. Instead of taking care of policies or users' identifiers, these approaches assume that communications are anonymous. They aim to hide users' true identity with respect to emitted location information.

A common way to hide the real location of users from the LBS provider is to use the K-anonymity property. This property addresses the conflict between information loss and disclosure risk [13].

K-anonymity and cloaking approaches have some limitations. First, by design cloaking relies on a trusted entity to make users' locations anonymous. Thus all queries should trust this entity during the system's normal mode of operation. The entity can also become a single point of failure; potentially creating a scalability bottle-neck because several handshakes must occur between the user and the entity to permit exchange of user profiles and anonymity measures. Another limitation of cloaking techniques is that either the quality of service or overall system performance decreases significantly as users choose to have more strict privacy preferences. For example, if a user requires better K-anonymity, the system needs to increase K for that user, which would result in a large cloaked area and hence less accurate query responses. Alternatively, if one requires to maintain the quality of service; the location server must resolve the spatial query for each and every point in the cloaked region and send the entire bulky result to the entity to be filtered out. This process clearly affects overall system performance, communication bandwidth, and server throughput and results in more sophisticated query processing. Finally, the concept of K-anonymity does not work in all scenarios. For example, in a less populated area, the size of the extended area can be prohibitively large in order to include K-1 other users.

Much work that has been based on cloaking can be found in the literature [11, 33,34]. One of the most recent advances in anonymity-based approaches was proposed in [35], which extends an early anonymity-based approach version [33]. This extension allows users to define their personal privacy requirements (i.e., the number K of users

among whom they wish to be anonymous) and the maximum delay and location perturbation they are willing to accept.

Another similar method called PrivacyGrid is described in [36]. Although the third-party entity described in [35] and the PrivacyGrid approach are very similar, the latter seems to be more efficient due to the cloaking techniques based on grids.

Mohamed et al. [37] tackle the privacy problem in a way that protects user privacy while keeping the functionality of LBSs. The main idea is to employ a third trusted party, called the Location anonymizer, which: (1) receives the exact point location from the mobile user, (2) blurs the location point into a cloaked spatial region according to certain constraints provided by the user, and (3) sends the cloaked spatial region to the location-based database server. Then, the location based database server is equipped with special modules that modify its functionality to work on the cloaked spatial region rather than on an exact point location. The penalty of having the blurred location information is that the location-based database server may not be able to provide a high quality service to its users. Users would have the ability to establish a set of parameters to balance the amount of information they would like to reveal about their locations and the quality of service they obtain from the location-based database server. Those parameters include 1) the level of anonymity (k), which users should specify their convenient level of privacy by introducing the anonymity parameter k, 2) the minimum area which represents the minimum area requirement of the cloaked spatial region, 3) the maximum area which represents the maximum area requirement for the cloaked spatial area, and 4) the temporal constraints that establish times during which these parameters apply. The

disadvantage of this approach is that there is a tradeoff between anonymity and accuracy of service. To adjust the accuracy, the user should be aware of different parameters. This in fact brings inconvenience in use of location based service if the user wants to find a best balance between privacy and accuracy for individual.

Gruteser and Grunwald. Present middleware architecture and algorithms that can be used by a centralized location broker service. The adaptive algorithms adjust the resolution of location information along spatial or temporal dimensions to meet specified anonymity constraints based on the entities that may be using location services within a given area. In their model, the mobile nodes communicate with external services through a central anonymity server that is part of the trusted computing base. In an initialization phase, the nodes set up an authenticated and encrypted connection with the anonymity server. When a mobile node sends position and time information to an external service, the anonymity server perturbs the position data according to a cloaking algorithm to reduce the re-identification risk. Moreover, the anonymity server acts as a mix-router, which randomly reorders messages from several mobile nodes, to prevent an adversary from linking ingoing and outgoing messages at the anonymity server. Finally, the anonymity server forwards the message to the external service. They consider a subject as k-anonymous with respect to location information, if and only if the location information presented is indistinguishable from the location information of at least k-1 other subjects. The key idea underlying the cloaking algorithm is that a given degree of anonymity can be maintained by decreasing the accuracy of the spatial data revealed. To

this end, the algorithm chooses a sufficiently large area, so that enough other subjects inhabit the area to satisfy the anonymity constraint. However, as the size of area increases to satisfy k-anonymity, accuracy decreases rapidly. Further, since there is an encrypted link between the anonymity server and the user nodes, this approach may also suffer from the disadvantages of encryption-based approaches.

Beresford and Stajano seek to make location information anonymous. Their approach requires that users are able to change pseudonyms repeatedly, even while they are being tracked. Users may adopt new applications with which they interact, but if the system's spatial and temporal resolution is sufficiently high, the applications can without difficulty link the old and new pseudonyms. The authors sought to address this problem by introducing the mix zone. An application doesn't receive any location information when users are in a mix zone, where identities are "mixed." Assuming users change to a new unused pseudonym whenever they enter a mix zone, applications that see a user emerging from the mix zone cannot distinguish that user from any other who was in the mix zone at the same time and cannot link people entering the mix zone with those leaving it. The infrastructure delays and reorders messages from subscribers within a mix zone to confuse observers. The problem with this system is that there must be enough subscribers in the mix zone to provide a suitable level of anonymity. In addition, if a mix zone has a diameter much larger than the distance the user can cover during one location update period, the system might not mix users adequately.

Lin et al. [38] noted that service providers can supply users with the information they need, they have to track user movements and location. In this paper they deal with such a problem by establishing a framework to protect location privacy. The main idea of their system is to forward transformed user location data to the service provider. They assume a number of different types of transformations, such as scaling, translation, or rotation, to hide user information. They use ***m*** numbers of agents interposed between users and service providers to implement these transformations, thus avoiding the negative effects of the single agent. For instance, if an enemy hacks one agent, it is still unable to track the user. If some agents illegally keep details about consumers, they will not be able to discover information about users without colluding with other objects. However, the problem with approach might be that the overhead for data transformation will increase with the increasing of number of transformation functions. Also this approach makes querying more complicated than ever because queries must be transformed in order to get a correct answer.

Trusted-agent and anonymity approaches do not scale up with the number of users in a system as well as encryption-based approach. This weakness is due to an intermediate layer known as the trusted agent that exists between the user and the service provider. The computing capability of this agent is limited; therefore, as the number of users in the system increases, the performance degrades due to a bottle-neck at the trusted agent.

Since every time the user gives a request, the trusted agent must satisfy the condition of k-anonymity, this involves overhead in processing in order to satisfy this condition. More essentially, whenever the users send information to the agent, the agent must use some cloaking function to hide the exact information. This will inevitably increase the computing load and degrading performance.

2.3. OBFUSCATION-BASED APPROACH

Obfuscation-based techniques are aimed at protecting location privacy by degrading the accuracy of the location information while still maintaining an explicit association with the real user identity.

Several techniques based on obfuscation have been proposed to preserve the privacy of users of context-aware services. Suppose that the service is not completely trusted by the user; therefore, since he considers his current activity (e.g. meeting customers) a sensitive information, whether to allow or deny the access to his precise current activity may be unsatisfactory. In fact, denying access to that data would determine the impossibility to take advantage of that service, whereas allowing access could result in a privacy violation. In this case, the only flexible solution is to obfuscate [39] the private data before communicating it to the service provider. Obfuscation techniques have been applied in the past to the protection of micro data released from databases (e.g., in [40]).

Various obfuscation-based techniques to control the release of location information have recently proposed (e.g., [41,42,43,44]) based on generalization or perturbation of the user's position. Semantic eWallet [45] is one of the first attempts to

support privacy in generic context-aware systems through obfuscation mechanisms. Users of the Semantic eWallet may express their preferences about the accuracy of their context data based on the requester's identity and on the context of the request. By abstraction, the user can choose to generalize the data provided, or to omit some details about it.

Other recently proposed obfuscation methods can be found in [43], which proposes that the real location of LBS users be replaced by circular areas of variable center and radius. One of the most recent proposals for non-collaborative trusted-third-party-location privacy is Space Twist [44]. This approach determines the point of interest closets to the real location, but the LBS provider cannot determine the real location of the user. The main advantages of this approach are that it requires no trusted third party or collaboration, and it hides the location of the user in a controlled area. However, this method is not able to achieve K-anonymity properties due to the lack of collaboration.

Obfuscation techniques aimed at location privacy protection. Location obfuscation is complementary to anonymity. In particular, rather than making users anonymous, obfuscation-based solutions assume the identification of users and introduce perturbations into collected locations to decrease their accuracy. Duckham and Kulik [41] developed an obfuscation technique to protect location privacy by artificially inserting into measurements some fake points with the same probability as the real user position. Their paper proposes a formal framework that provides a mechanism to balance user needs for high-quality information services with location privacy.

Obfuscation has several important advantages that complement other privacy protection strategies. Obfuscation and anonymity are similar in that both strategies attempt to hide data in order to protect privacy. The crucial difference between them is that whereas anonymity aims to hide a person's identity, obfuscation is an explicitly spatial approach to location privacy that aims to allow a person's identity to be revealed. Potentially, this combats one of the key limitations of anonymity approaches: the need to authenticate users. At the same time, degrading the quality of location information makes inferring identity from location more difficult. If flexible enough to be tailored to specific user requirements and contexts, obfuscation (unlike regulatory strategies) does not require high levels of complex infrastructure and (unlike privacy policies) is less vulnerable to inadvertent disclosure of personal information. Unlike many anonymity approaches, it is lightweight enough to be used without the need for trusted privacy brokers.

Obfuscation seeks a balance between the level of privacy of personal information and the quality of the LBS. Recent research has indicated that there exist many situations in which it is possible to expect high-quality LBSs based on quality positional information. Consequently, in situation where the user requires a higher quality of service than can be achieved at a user's minimum acceptable level of privacy, then other privacy protection strategies must be relied upon instead. Moreover, obfuscation assumes that the individual is able to choose what information about his or her location to reveal to a service provider. Although this assumption may be realistic when using client-based or network-assisted positioning systems and when sharing location information with a third-

party location-based service provider, dealing with the entities that administer network-based positioning systems still requires privacy protection based on regulations.

2.4. POLICY-BASED ENCRYPTION APPROACH

Traditional public key encryption is coarse grained: A sender encrypts a message M with respect to a public key, and only the owner of the (unique) secret key associated with the public key can decrypt the resulting cipher text and recover the message. These straightforward semantics suffice for point-to-point communication, in which encrypted data is intended for one particular recipient who is known in advance to the sender. In other settings, however, the sender may instead wish to define a policy determining who is allowed to recover the encrypted data. For example, classified data might be associated with certain keywords to which only certain users have access.

One of the first solutions was proposed in [46].The authors propose a form of encrypted query processing combining the use of a data structure suited for managing spatial information with a cryptographic schema for the secret sharing. On the server side, location data are handled through a directed acyclic graph (DAG), whose nodes correspond to Voronoi regions obtained by a tessellation of the space.

Recently, [47] proposed a cryptographic approach inspired by the private information retrieval (PIR) fiel. The service provider builds a Voronoi tessellation according to the stored pois, and superimposes on it a regular grid of arbitrary granularity. Some of the advantages of this approach are that location data are never disclosed; the user's identity is hidden among identities of all users. However, since mobile devices are often characterized by limited computational capability, the query

encryption and answer processing performed on the client side have a strong impact on service response time, and power consumption.

A cryptography system with a single secret or shared key requires users to distribute the key securely before they can communicate in private. Such distribution can be complex, especially if the number of users is very large.

Zhong et al. [48] designed novel protocols to implement location-based services for mobile wireless users without using a trusted third party. One allowed users to control which entities can have access to their location information. They identify two major types of LBS: The first type of services directly transfers users' location information to authorized entities and that protect the information from unauthorized entities, including the service provider itself. The second requires computations that take user locations as inputs. The challenge is how to complete these computations without revealing user locations. The basic idea of their privacy-preserving design is that only the entities in the authorized subset should be able to derive the key to decrypt the location information. They have shown that their protocols have low overheads and are suitable for personal mobile devices.

A policy-based encryption method has the advantage that encryption and access control are in one package. Not only would it provides for encryption during transmission, but also for storage. Thus ensuring that all sensitive information being transmitted is secure. The encryption scheme also allows us to implement access control as only those who satisfy the policy can decrypt and have access to the record. There is

no need for users to manage their own keys nor is there need to distribute keys. There will be no difference for the user experience between the two choices. In both cases, users log in and gain access to documents for which they have permission.

A common disadvantage of any encryption scheme is that records are stored encrypted. The decryption keys must be kept for the lifetime of the record. The decryption algorithm must also be available to recover the original documents. In order to ensure these are both available, it may be necessary to keep a backup of the keys and algorithm. Otherwise, it would be possible to store the records in readable format somewhere physically secure and not connected to a network. We would need to store backup copies of the records in plaintext regardless of which system were used, due to possibility of hardware failure.

3. LOCATION PRIVACY POLICY MANAGEMENT SYSTEM

This section will first present a succinct yet expressive location privacy policy language. Then, it will describe the algorithms for policy conflict detection and policy composition. Finally, there will be an explanation of the policy recommendation system.

3. 1. DEFINTION OF LOCATION PRIVACY POLICY

This work particularly focus on location privacy; which can be define as the ability to prevent other unauthorized parties from learning one's current or past location. Polices can be broken into three categories: specialized, generalized and exception. The three policies interact with one another and the results determine whether a user is visible to his relationships or not. Each of the three basic policies work in very much the same way, with a few small differences.

Definition 1: Let U1 be a creator of a policy, let U2 be the target of the policy. Let Pe be a policy where e is the type of policy: specialized, generalized, or group privacy. A policy will contain the following, a time period Ttime consisting of a period in hours, and a range of days; a range of location, Location policy, that can consist of location, city, state and country.

A few format conditions apply to all three policies. First, the targeted user or relationship must be part of the database before a policy is valid; similarly, a user may only make policies for people on his relationship list. The logic behind this rule is that if the users want to create a policy, they must expect that the policy will be shown. Further, if a user allows another user to see him, he must have some form of relationship with that other user. Another common condition deals with time; at least 15 minutes must elapse

between the start and end time of any of the tree policies. It is assumed that any a considerable amount is need for a policy to be valid. Third, location must be fully defined for a policy; that is; if a user enters information in the state filed, he must also have provided the information for the country line, just as a state must be entered if a city is entered and the same for location and city.

3.1.1 Specialized Policy. Specialized Policies are the main type of policy; they require a connection between the creator of the policy, and the targeted user. Once a policy is created for a target, whenever the target requests a list of all visible users during the allotted time for the policy; the creator is displayed in the target's report. Specialized policies consist of the creator; the target; the location of the policy, and the time and day of the policy. For example, if the creator wants to let the target to know that he will be at the bank on Monday, he simply enters the location (Bank, City, State, and Country), the target , the time (Stime - Etime), and Monday for both start and end date.

Definition 2: Let U1 be the creator user ,and let U2 be the individual target. Any policy created by U1 for U2, P, will contain a period of time Ttime and a range of location RangeLocation. During the time period Ttime the range of location RangeLocation will be reported to U2 as the location of U1.

Checking for Specialized Policy:

Let U1 be requesting user, Let U2 be owner of Pi
Let t be time of request
Let T be the Pi period of time
Let Rloc be location of Pi
Let Pt be policy type

```
Let Si be the ith policy of type Pt whose target is U1

    For each policy in S, i
        If t is within Si.T
            Print U2 to screen
            Print Rloc to screen
    Loop
```

3.1.2 Generalized Policy. Generalized Policies, on the other hand, pertain to the connection between the creator of the policy and a relationship role. A user can assign each of his relationships to one of three roles, each one general enough to include most of his associates: family, friends, and colleagues. Each user within a relationship role is subject to the policy defined by the user for the role. Like a specialized policy, a generalized policy consists of creator, a time period, and location. It differs in that instead of identifying a specific user as target, the creator chooses one of the relationship types for the policies. Generalized policies also have allowed the user to specify a privacy level for the policy. For example, let U1 want to tell his friends that he is at school all this week from 7 am to 3 pm. So he creates a group policy with time range of 7:00 to 15:00, Monday to Friday, with a range of location of School, Rolla, MO, USA. U1 sends this policy to be validated, if no policy conflicts, it will then recommend a privacy level for the policy. Because of the average time span and very specific location, as it will be described later on group privacy policy. The privacy level of the policy is set to high and established in the database.

Definition 3: Let U1 be the creator user, let Re be the targeted role where e is a role (family, friend, or colleague). Any policy, p, created by U1 for Re, P, will contain a period of time Ttime and a range of location RangeLocation. During the time period Ttime the range of location RangeLocation will be reported to U2, where U2 is a relation of U1 within the category e, as the location of U1.

Checking for Generalized Policy

Let U1 be requesting user, Let U2 be owner of Pi
Let t be time of request
Let T be the Pi period of time
Let Rloc be location of Pi
Let Pt be policy type
Let Si be the ith policy of type Pt

For each policy in S, i
    Let Ro be the target role of policy Si
Let Zj be the jth user in a search of users that reside in Si.U2's relation list with role Ro
For each user in Z, j
If U1 equals Zj and t is within Si.T and Z.Privacy Level greater than or equal Si.Privacy level
Let Qn be the nth exception with target U1 and owner is Si.U2
For each exception in Q, n
        If t is within Qn.T
            Ignore
Else
Print U2
            Print Rloc
    Loop

3.1.3 Exception Policy. Exceptions are additions to group policies; they exclude a specific user from a group policy. In other words, the policy creator removes his location from any reports that would be visible to the targeted user. Exception policies are also much like individual policies, however, because their function is to hide information from a specific user; they require no location. Exceptions can be set up to allow the policy creator to hide his location from specific user for part or all of a group policy time period. They also allow the exception policy to overlap two or more policies. Going back to the generalized policy, assume that the creator of the policy is shopping for a gift for one of his family member, so he may create an exception policy for this individual.

Definition 4: Let U1 be the creator user, and let U2 be the individual target. Any exceptions, e, created by U1 for U2, will contain a period of time, Ttime. If U1 has a generalized or group privacy definition during the time Ttime, the location of U1 will not be reported to U2, regardless of the definition's targeted role.

3.1.4 Group Privacy Policy. This policy differs from the other three in that it is three policies in one. This type of policy allows a user to display his location to all users that are assigned to a certain role, but to limit the information available to each user in a given role based on privacy level. A group privacy policy contains the creator, the targeted role, and a time period and location range for each of the three levels of privacy.

For example, user1 wants to create a policy to tell all his friends that he will be in Montana for the weekend, but he only wants to let his closet friends know what city he will be in. So he sets the target role to friend, and enters a time period and location range

for each level of privacy. For close friends, User1 wants his friends to know that from Friday to Sunday All day (00:00 – 23:59), that he will be in Helena, Montana, USA. For most friends, he only wants them to know that he will be in the state of Montana; therefore, he specifies the same time period, but changes location range to Montana, USA. Lastly for friends who are only allowed to know that he will be on his way home at 8:00 am on Monday, he changes the time range to Monday from 5:00 to 8:00 am and the location range to Airport, Helena, Montana, USA.

Checking for Group Privacy Policy

```
Let U1 be requesting user, Let U2 be owner of Pi
Let t be time of request
Let T be the Pi period of time
Let Rloc be location of Pi
Let Pt be policy type
Let Si be the ith policy of type Pt
For each policy in S, i
        Let Ro be be target role of policy Si
Let Zj be the jth user in a search of users that reside in Si.U2's relation list with role
Ro
For each user in Z, j
If U1 equals Zj and t is within Si.T and Zi.Privacy Level equal Si.Privacy level
Let Qn be the nth exception with target U1 and owner is Si.U2
For each exception in Q, n
If t is within Qn.T
                                Ignore
Else
                                Print U2
                                Print Rloc
```

Loop
Loop

3.2. POLICY CONFLICT DETECTION

All forms of conflict deal with similar elements and terms. The basic idea to conflict is that User1 cannot be in two places at once; therefore a conflict can occur for either time, location, or both. Location levels are based on a location hierarchy, figure (2). Location is the most specific level, and country is the least. The location level that takes priority depending on the type of interaction.

The concept of similar times is tricky because hours must be defined as circular time and weekdays as linear. Let Se be the start time for the established policies, Sn be the new policy start time, and Ee and En be the established end time and new end time, respectively. Two policies have similar times if either the start or end time of the new policy is between the established start and end time as shown in Figure 3.1.

Figure 3.1 The location hierarchy for the system.

Two policies also share similar time if the established time period of one is between the new time periods as shown in Figure 3.2.

Finally, because of the linear timeline, if the established end time is less than the established start time, we need to see if either the new start time or the new end time is between the established start time and one second before midnight, 23:59:59, or midnight 00:00:00 and the established end time as shown in Figure 3.3.

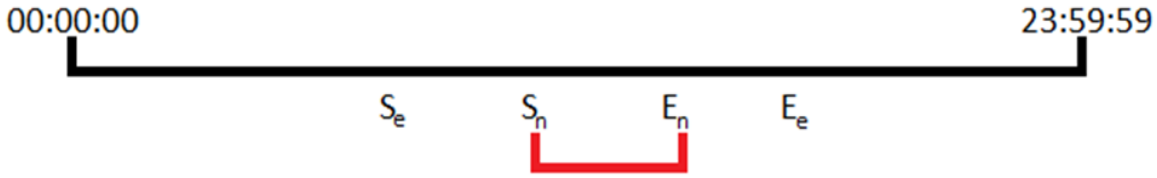

Figure 3.2 Time conflict.

(Se ≤ Sn ≤ Ee) OR (Se ≤ En ≤ Ee).

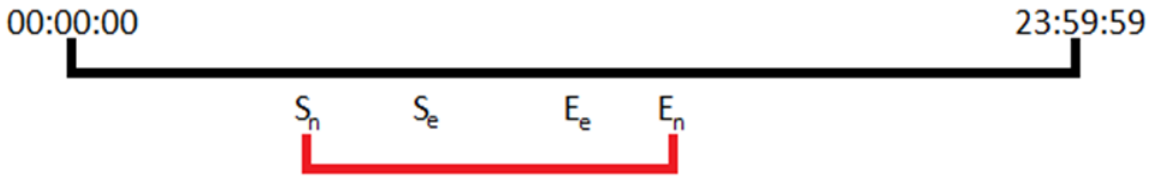

Figure 3.3 Time conflict.

(Sn ≤ Se) AND (Ee≤ En).

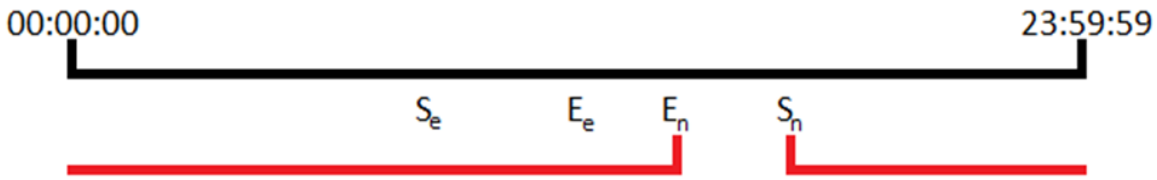

Figure 3.4 The circular time line conflict.

(EE ≤ SE) AND ((SE ≤ (SN or EN) ≤ 23 : 59 : 59) OR (00 : 00 : 00 ≤ (SN or EN) ≤ EE)).

Let Se and Ee be the time period of an established policy, and let Sn and En be the time period for new policy. If Se is 3:00 and Ee is 13:00, neither Sn nor En can be between the established time nor can the Sn start before Se or the En ended after Ee. If one were to enter Sn as 17:00 and En as 16:00, it would conflict with the established time, as the time periods overlap

3.2.1. Types of Conflict.

Generalized-to-Specialized Policy

Not all generalized and specialized policies conflict. Conflict requires that specific conditions must be met. First there must be a similar time and day range. A generalized policy will conflict with established specialized policy if it has the same location level or higher. Inversely a specialized definition conflicts with an establish group if it's location level is the same or lower than the established generalized definition. For instance, Let there an established generalized policy by U1 that is a middle level policy, and let user have three friends; Uh, Um, and Ul; that have a high, middle, and low privacy level respectively. U1 tries to create a policy for each of his friends that overlap

in time with the established group policy. Based on our assumption the established generalized will conflict with the policy of the high and middle level users, but as Ul cannot see the generalized policy, it will not conflict with the generalized policy; like in the figure below. However, if the generalized policy is changed to a low level policy, it will conflict with all three users, as all users can see the policy.

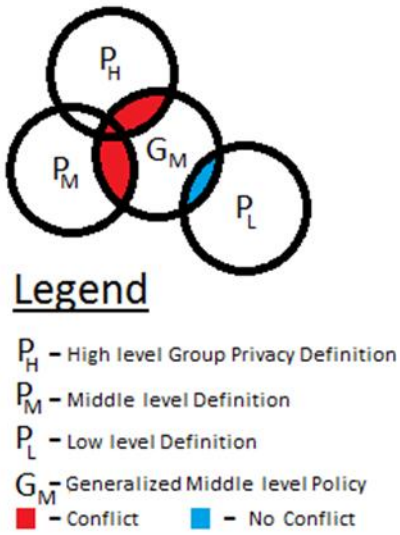

Figure 3.5 Conflict between generalized to specialized policy.

Conflict Algorithm Generalized to Specialized:

Let Pt be policy type
Let U1 be creator, Let U2 be target
Let T be period of time
Let Rloc be range of location
Let Ro be role type as found by search of relation list of U1 for target U2
Let Z be pending policy
Let Si be the ith policy in a search of policy type Pt whose owner is U1
For each policy in S, i
Let Ro2 be role type as found by search of relation list of U1 for target Si.U2

If Si.T over laps Z.T and Ro equals Ro2

If Si.Rloc less than or equal to Z.Rloc or Z.Rloc in different location from

Si.Rloc

Conflict Algorithm Specialized to Group Privacy Policy

Let Pt be policy type

Let U1 be creator, Let U2 be target

Let T be period of time

Let Rloc be range of location

Let Ro be role type as found by search of relation list of U1 for target U2

Let PL be privacy level as found by search of relation list of U1 for target U2

Let Z be pending policy

Let Si be the ith policy in a search of policy type Pt whose owner is U1 and target role is Ro and Privacy level is PL

For each policy in S, i

If Si.T over laps Z.T

If Si.Rloc greater than or equal to or Z.Rloc in different location from Si.Rloc

Notify of conflict

Loop

Generalized-to-Exception Policy

Generalized and exception policies do not conflict because the latter are part of the former. In other words, exceptions are used in conjunction with generalized policies and must therefore be able to overlap one another.

Another way to think of exceptions is as a subset for any generalized policies that occur at the same time. As a subset of the generalized policies as presented in the figure below.

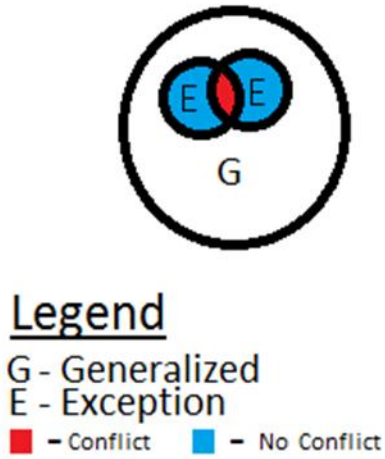

Figure 3.6 Overlapping between the generalized and exception policy.

Specialized-to-Exception Policy

Because a specialized policy makes a user visible, and exception policies effectively do the opposite, any overlap between the two is considered a conflict. A User1 cannot request that he be both visible and invisible.

Group-Privacy-to-Generalized policy

A group privacy policy is essentially three group policies connected in a single policy. As such, the conflict between a generalized policy and a group privacy policy may exist between any of the three group policies individually and the generalized policy.

If any of the times conflict, then both policies are in conflict with one another; see Figure 3.7.

Conflict Algorithm Group Privacy Policy to Generalized

Let U1 be the creator, Let U2 be the target
Let T be the time period
Let Rloc be the range of location
Let Z be the Pending Policy
Let Si be the ith policy of a search of policy type PG owned by U1 with target U2
For each policy in S, i
    Let E be Z.PPe
    If E.T over laps Si.T and E.Privacy Level is less than or equal to Si.Privacy Level
        Notify of conflict

Conflict Algorithm Generalized to Group Privacy Policy

Let U1 be the creator, Let U2 be the target
Let T be the time period
Let Rloc be the range of location
Let Z be the Pending Policy
Let Si be the ith policy of a search of policy type PP owned by U1 with target U2

For each policy in S, i
    Let E be Si.PPe
    If Z.T over laps E.T and Z.Privacy Level is greater than or equal to Si. Privacy Level
        Notify of conflict
Loop

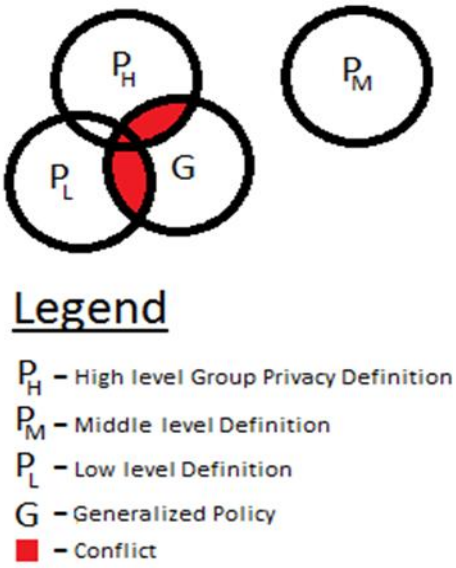

Figure 3.7 Conflict between the generalized and group privacy policy.

Group Privacy-to-Exception

Exception policies interact with group privacy policies in much the same way as they interact with group policies. Exceptions cannot conflict with group privacy policies because they are used with such parties.

Group-Privacy-to-specialized

Unlike generalized definition, specialized definitions will not conflict with an entire Group Privacy definition, but with the individual policy that corresponds to the target user's privacy level. Take for instance, that the example described for group privacy definition was enacted. If the creator of the group privacy policy, User1, wanted to create an individual policy for a low privacy level friend at 4:30 am – 5:30 am Monday for the Hotel Lobby, the new individual policy would conflict with the lower level of the group privacy policy, but if the day of the week were to be changed to Sunday, which would conflict the middle and high level parts of the policy. However, even if we made

this change as the targeted friend is a low level user, the two policies do not conflict, and the policy is allowed to be established as shown in Figure 3.8.

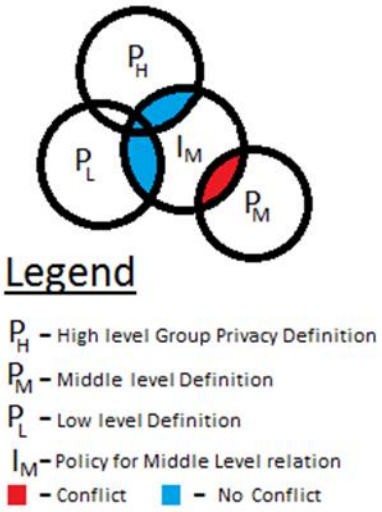

Figure 3.8 Conflict between Group-Privacy and specialized.

Conflict Algorithm Group-Privacy Policy to Specialized Policy.

Let Pt be policy type
Let U1 be creator, Let U2 be target
Let T be period of time
Let Rloc be range of location
Let Ro be role type as found by search of relation list of U1 for target U2
Let PL be privacy level as found by search of relation list of U1 for target U2
Let Z be pending policy
Let Si be the ith policy in a search of policy type Pt whose owner is U1
For each policy in S, i
    Let Ro2 be role type as found by search of relation list of U1 for target Si.U2
    Let PL2 be privacy level as found by search of relation list of U1 for target Si.U2
    If Si.T over laps Z.T and Ro equals Ro2 and PL2 equals PL
        If Si.Rloc less than or equal to or Z.Rloc in different location from Si.Rloc
            Notify of conflict
Loop

Conflict Algorithm Specialized to Group Privacy Policy

Let Pt be policy type
Let U1 be creator, Let U2 be target
Let T be period of time
Let Rloc be range of location
Let Ro be role type as found by search of relation list of U1 for target U2
Let PL be privacy level as found by search of relation list of U1 for target U2
Let Z be pending policy
Let Si be the ith policy in a search of policy type Pt whose owner is U1 and target role is Ro and Privacy level is PL
For each policy in S, i
If Si.T over laps Z.T
If Si.Rloc greater than or equal to or Z.Rloc in different location from Si.Rloc
Notify of conflict
Loop
Notify of conflict clear

3.2.2 Outline of Policy Conflict Detection Algorithm. When checking for the location of other users, the system first differentiates between the three types of visible definitions. The tables for each type of policy are requested individually from the database. Policies are separated by the targeted users and roles, after separating the policies, they are selected by whether the selected user is within the group of the current policy. The policies are then selected if they are within the time of the policies are selected. Finally, the policies are compared to established exceptions, if they conflict, then the conflicting policy is not retrieved.

3.3. POLICY COMPOSITION

3.3.1. Policy Insertion. To insert a policy into the system is to create a new policy to be inserted into the database. This is the most used function of the program, to be able to add new policies. Policies are first submitted, then are checked for validity, then are checked against established policies to confirm or deny conflicts and then lastly if the policy is valid, it is inserted into the database.

3.3.2. Policy Modification. Modify polices is more or less altering established policies in accordance to conflict conditions. Much like inserting the user resubmits the policies, which in turn is check for validity, compared to other established policy, and finally reemitted to the database.

3.3.3. Policy Deletion. Deleting a policy is straight forward. When a policy is requested to be deleted, the policy is first checked if it has multiple definitions, that is to say multiple time periods. If there does exists multiple definitions, only the requested definition is to be deleted, on the other hand if there is only one definition, the entire policy is deleted.

3.3.4 Merging Policies. When an established policy is merged with a new policy, the new time range of the policy is added to the existing policy, allowing a policy to be enacted at multiple times. The standard conditions for such actions are simple: The two polices must share a targeted user, creator, and location, and they must not overlap in time. There are added conditions, however, for each of the four policy definitions. For specialized and generalized definitions, there are no added conditions if the above conditions are satisfied and the creator wishes to merge the policy, they will merge.

Exceptions have no merging conditions as exceptions cannot be merged as they have no information to show to a user. Group Privacy definitions are a little different, for merging to be possible for a group privacy policy, the above conditions must be held true for all three policies.

Merging Policy

Let Pt be policy type
Let U1 be creator of policy, Let U2 be target
Let T be period of Time
Let R_{loc} be range of location
Let Z be pending policy
Let S_i be the i^{th} policy from a search of policies of type P_t with creator U_1

For each policy in S, i
If $Z.R_{loc}$ equals $S_i.R_{loc}$ and Z.T does not overlap $S_i.T$
    Request for merge
    If request is accepted
        Create merge policy
    Else
    Clear Z
        Loop to beginning

3.4. POLICY RECOMMENDATION SYSTEM

The policy recommendation system will generate recommended policies based on users' basic requirements in order to reduce users' burden. In what follows, we first define a concept of privacy level and then present our recommendation algorithm.

3.4.1 Definition of Policy Privacy Levels. When a user creates a new group policy, they are recommended a privacy level for said policy. This privacy level dictates which categories of relations in a role are allowed to see the policy. Visibility is based on a level hierarchy, with High privacy at the top and Low privacy at the bottom. Any user at the group policy's privacy level or high can see the policy, that is low level policy can be view by all three level, high and middle level friends can see middle level policy and only high level friends can see high level policies. Privacy levels for group policies are based on a calculated number, which is created from an algorithm based on the range of location and period of time. The user designates in a separate window the location weight, that is to say what location level policy weight is, like making the weight of City level policy to be 15 and Location level to 0. The default weight for each location level 0 for location, 20 for City, 55 for state and 90 for country, as each is more generalized then the next. The Location weight is subtracted from the time weight, which can be between 0 and 100 to calculate the privacy level, as show below:

$$Privacy\ Level = \frac{|S_{time} - E_{time}|^{.613}}{7 - |S_{day} - E_{day}|} * 100 - Location\ Weight$$

Stime is the start time for a policy hours.

Etime is the end time for a policy hours.

Sday Start time for a policy day and Eday the end time for a policy day. Where the time line for the day start at Saturday=6, and end at Sunday=0.

The formula for time is set up so that is follows a Cartesian path similar to y= √x where y is the difference in time, and x is the difference in days. But the ratio between the

two differences must equal one when they are at their maximums, that is to say $23= 7^{1/x}$ and solving for x leads to x equal to .613. The ratio is then times 100 to turn the decimal into a percentage. After finding the privacy weight, if the weight falls between the cut off points for the three levels of privacy, that level of privacy is recommended. If the weight falls between 100 and 71it is a high level, if it between 70 and 24 it's middle and anything 23 or lower is a low level policy.

The cut off points are based on policy disturbed and the weights of generic scenarios of policies. For example a generic low level policy is 4 hours and 4 days and a state level policy, so its weight is $(4^{.613}/ (7-3)) * 100 – 55 \approx 3$, and a generic middle level is 8 hours, one day, and a city level policy weight is $(8^{.613}/ (7-0)) * 100 – 20 \approx 31$. But we also want to have the policy distributed so that a majority of them are low level policies, and that there are more middle level policies then high level. As such the middle level range needs to be expanded so that the percent distribution is 62.5% low policies, 23% middle policies, and 14.5% high level polices.

The default location weights for the four level of location; location, city, state and country; were decided based on estimates and less on a mathematical project. The weight for each location level does not need to be very exact, as the defaults are only to be used as a suggestion. As such the estimation are pretty loose location being zero as it needs to be the lightest and country needs to be the heaviest, so make things much simpler, the default is 100. As state level should be about between the location and country location add much closer to country, so making state level fifty-five leads to be closer to the country level and be about the middle of the spectrum. Lastly the city level must be between the state level of fifty-five and the location level of zero, and be closer to the

lighter location level, so I estimated the city level to have a weight of twenty. The reason for the large gap between the city and state is because there is a large difference between a city area and a state area, as such there should be a large gap between the two location levels.

3.4.2 Policy Recommendation Algorithm. The policy recommendation works as follows. Whenever a policy is inserted into the system, its privacy level will be computed and stored. Then, when a user would like to compose a new policy, the user can tell the system that what kind of policy that he wants. For example, the user can indicate that he would like to assign a medium level privacy policy to his new friend Bob. Upon the user's request, the system will search the existing policies and find the policies with medium privacy level. Among the retrieved policies, the policy which contains similar role of Bob will be returned to the user as a recommended policy. The user can either accept the recommended policy as it is or modify the policy. In this way, users no longer need to scribe new policies each time.

Recommend Policy

Let U1 be creator, Let U2 be targeted role
Let TS be starting hour, Let TE be ending hour
Let DS be starting day, Let DE be ending day
Let Rloc be range of location
Let S be policy in weight table with owner U1 and target role U2
For each element in S that matches range of Rloc, E
    Let location weight equal to E
    *Let Time weight equal to (|T_S-T_E |^.613/(|D_S-D_E |+1))*10*
    Let Privacy equal the absolute difference between location weight and time

```
weight
    If Privacy greater then 70
        Recommend High level in dialog box
    Else If Privacy greater then 21
        Recommend Middle level in dialog box
    Else
        Recommend Low level in dialog box
    Insert Policy in system with insertion algorithm with recommended level.
```

4.PERFORMANCE STUDY

The system model was implemented using Microsoft Access, this choice ensure ease of access, but leads to sacrifices some security features. The maximum file size of an Access database file is two gigabytes. In addition to a maximum size, there are also a maximum number of objects that can be within a database file within a given time. 32,768 objects can be part of the database as at any time. Lastly is the capacity of users that can be concurrently on the stream which is 255, but like file size one can split database across a distributed network of databases, effectively raising the cap on concurrent users.

The system was implemented based on two languages: Visual Basic and SQL.

- **Visual Basic:** As this experiment called for simply a model to test with, visual basic was an acceptable language to program end. By using visual basic we could create the need input and windows for the program without extra work, allowing more time for the internals of the program. Visual basic allowed me to quickly mock up the interface, and then add the needed underling code quickly and concisely. Visual Basic was also considered because I have had the experience needed to connect the user interface to the background database in SQL.

Visual Basic's function in the program was for the most part for the user interface. The interfaced needed to be simple, user friendly, and able to work correctly. The interface desired to guide a user to the proper policy and display the policies that user has already created. Visual Basic other function was to communicate with the database

using SQL and using the data giving to back to visual basic, evaluate whether a policy conflicts with established policies.

- **Structured Query Language**: SQL was planned to be used from the beginning to be used in the project as it is optimized for select statements and for creating and modifying both tables and tuples in the database used. Microsoft Access SQL was selected for ease of use. Although difficult to secure and not recommended for a complete system. It was adequate for this experiment, and it offered the added benefit of a graphical user interface, so that a table and database can be created easily and one can modify tuples quite quickly.

The primary function of SQL in the project was to run the database communications sent by Visual Basic. Because SQL is optimized for selecting tuples from tables using varying forms of condition, it is very vital to the functioning of conflict algorithms, as the conflict are very specific and among the four types of policies and their interaction with one another.

The following discussion evaluates the system in terms of both effectiveness and efficiency.

4.1 EFFECTIVENESS

The user interface was designed to be convenient and simple for the user. All user functions were set up to be in one place so that a user can easily maneuver to want they wish to do quickly. All the functions are grouped together in terms of what they affect. The interface is simplified so that it is easy to understand.

4.1.1 Main Window. The main window works as a hub for all processes the user wishes to do, it contains access to adding, modifying, and deleting both policies and relationships. The status window displays the user's name and the large area within the window, displays policies that the logged in users has in the system, displaying the targeted role or user, the location, and the start/end time of the policy, as shown in Figure 4.1. If a user wishes to see where other users are, he must click the See Relations button to bring up the visible relationship list. Exceptions are separated from the three other types of policy definition for clarity, and by clicking the exception window brings up a separate window that allows the users to add, delete and modify exceptions.

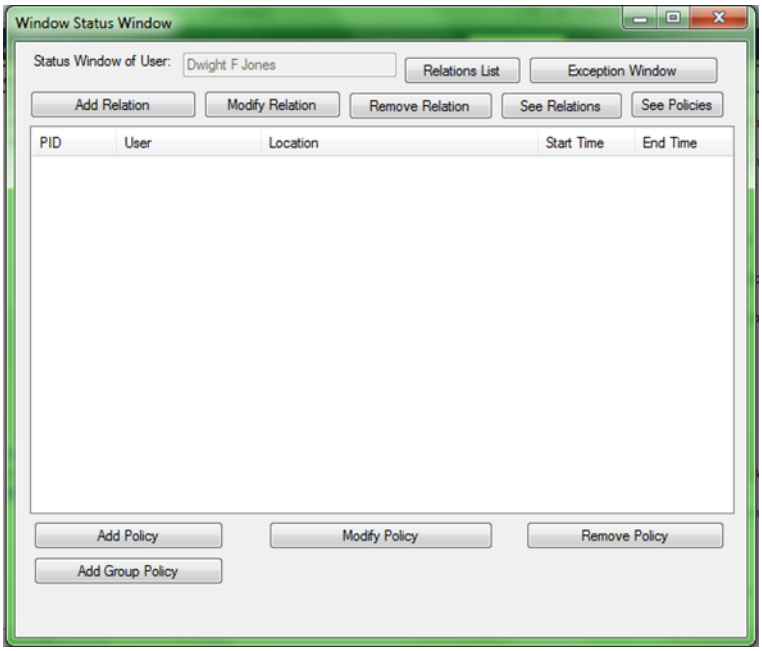

Figure 4.1 Main window of the system.

4.1.2 Adding Specialized or Generalized Policy. Clicking the Add Policy brings up a second window, and enter the user's ID into the first text box. The second row contains a dropdown box, this contain a list of all users with individual's relationship list. This makes it convenient for the individual to target a user, without having to remember the targeted user's ID. Figure 4.2 illustrates the range of location is separated into textboxes to better clarify what is to be entered. The start time and end time portion is done as a pair of numerical up and down, allowing the user to select time. Lastly the weekday appears as a drop down window for easy selection. The Adding group policies does not differ much from individual policy, except rather than a target user, there is a targeted role.

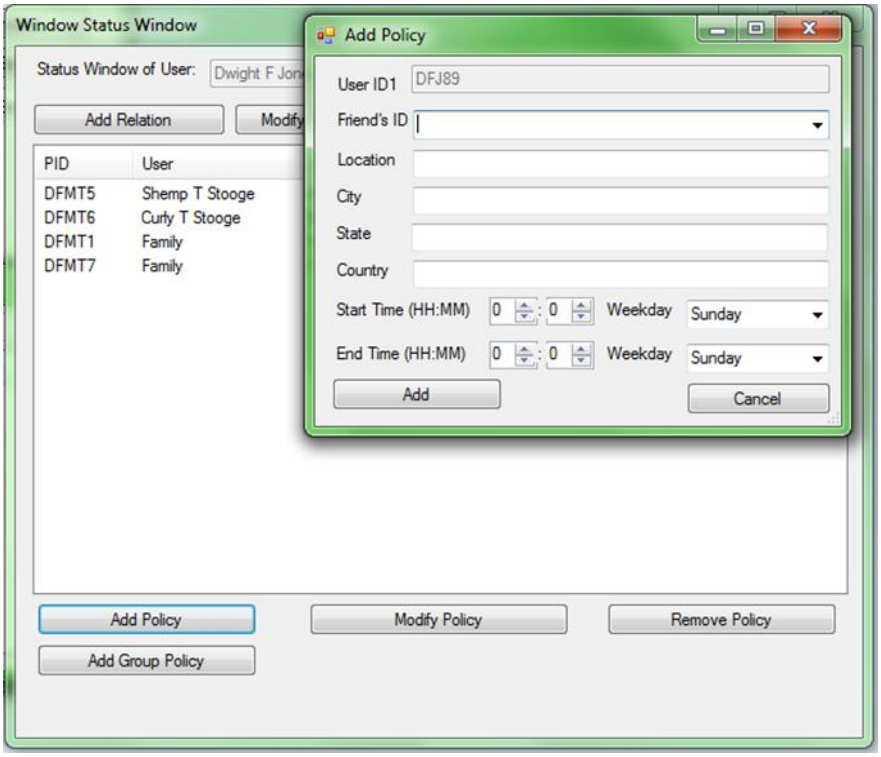

Figure 4.2 Adding specialized and generalized policy.

4.1.3 Adding Group Privacy Policies. Clicking the Privacy Policy button lead to the group privacy policy; see Figure 4.3. Like group and individual policy the policy starts with the creator's ID and a targeted role, but the elements contain change from here. Three buttons below the target role, these are the separate policy for each privacy level. The user clicks and input each level sparely clicking add, to close the window. After adding information for each level, clicking add in the main window to adds this policy to the user's policies. Before adding the policy to the database the system recommends a privacy level for the policy bring up a window with the recommend level and three button one for each of the level. After choosing the desired level, the policy is finally set into the database.

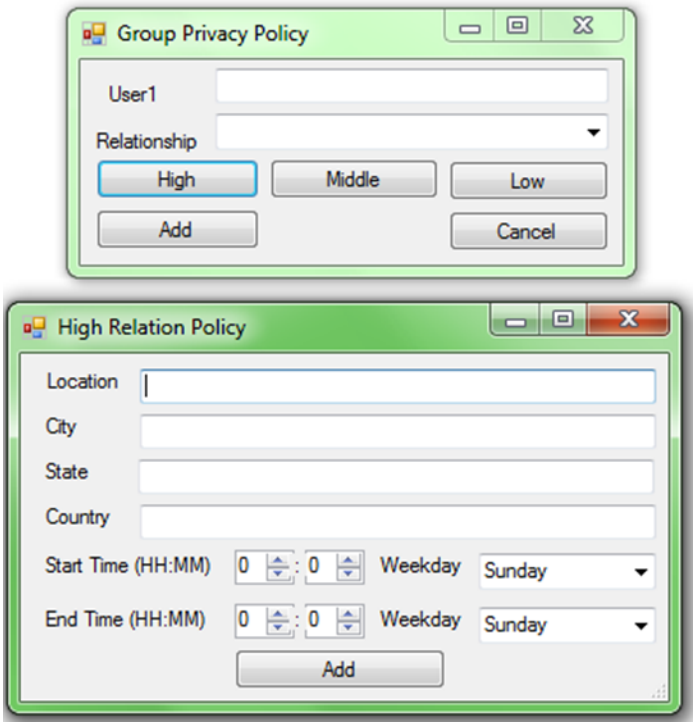

Figure 4.3 Adding group privacy policy.

4.1.4 Modifying Policies. By selecting an established policy then clicking Modify Policy, this bring up the modify policy window. Each window looks much like it respective add policy, except that the input boxes are filled with the previously entered input as shown in Figure 4.4. Changing any information within this window will change the information of the established policy.

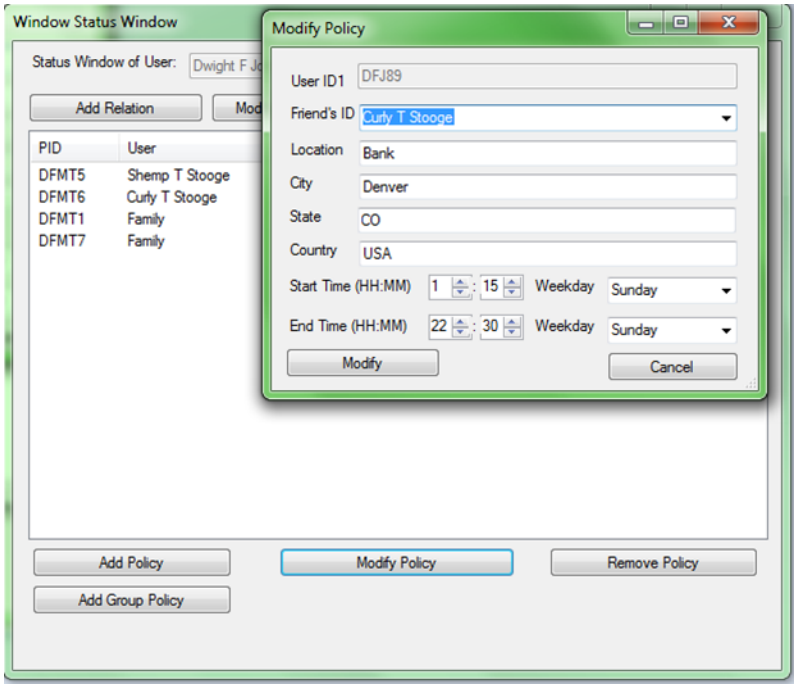

Figure 4.4 Modifying policy.

4.1.5 Deleting Policies. Deleting is very simple, first highlighting the established policy and clicks *Removing Policy*. This will bring up a confirmation dialog box, clicking yes will delete the established policy, while clicking no will return the user to the main status window. After clicking yes the main policy window will repopulate, showing that the policy has been deleted.

4.1.6. Exception Window. Clicking the exception window brings up *the exception status window*. The exception is separated from the other three policies to clarify the distinction between them. Aside from the separation, there is no really difference in appearance. This window contains the options for exceptions: adding, modifying, and deleting. When bringing up the window it populates with established exceptions. Like the main window the large area in the middle of the exception window, this is the policy window. This area displays established exceptions showing the policy's ID, the user that exception targets and the starting and ending time of the exception. Closing the window brings the main status window as illustrated in Figure 4.5.

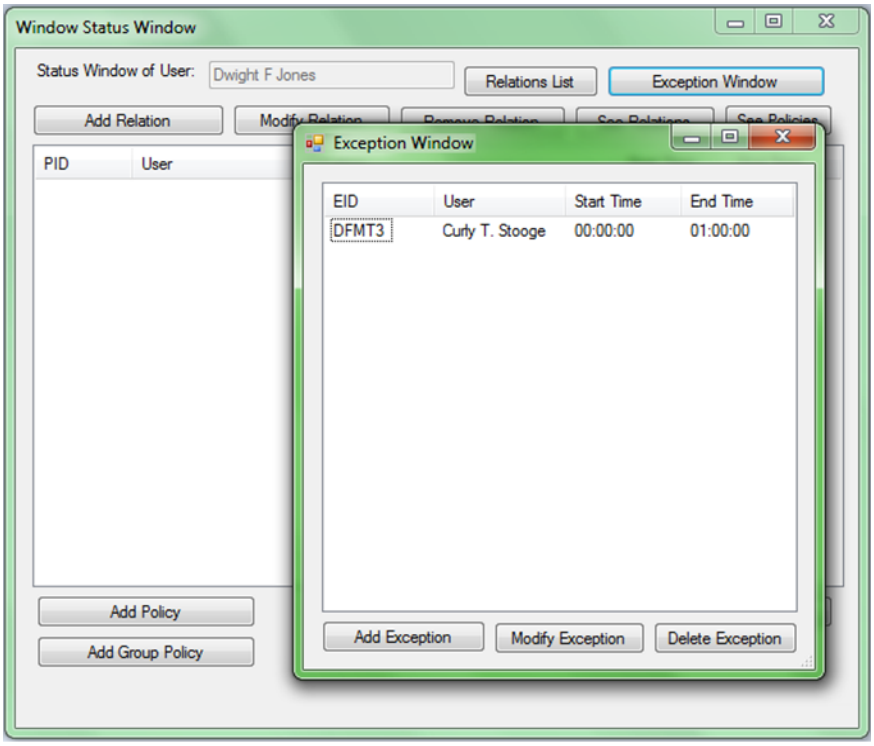

Figure 4.5 Show how to make exception policy.

4.1.7 Relationship Feature. The relationship functions are not important part of the system but are needed for policy definitions. The relationship list displays the user's relations and what role they are part of. Adding, modifying and deleting friend is much like how it is done with policy definition; see Figure 4.6.

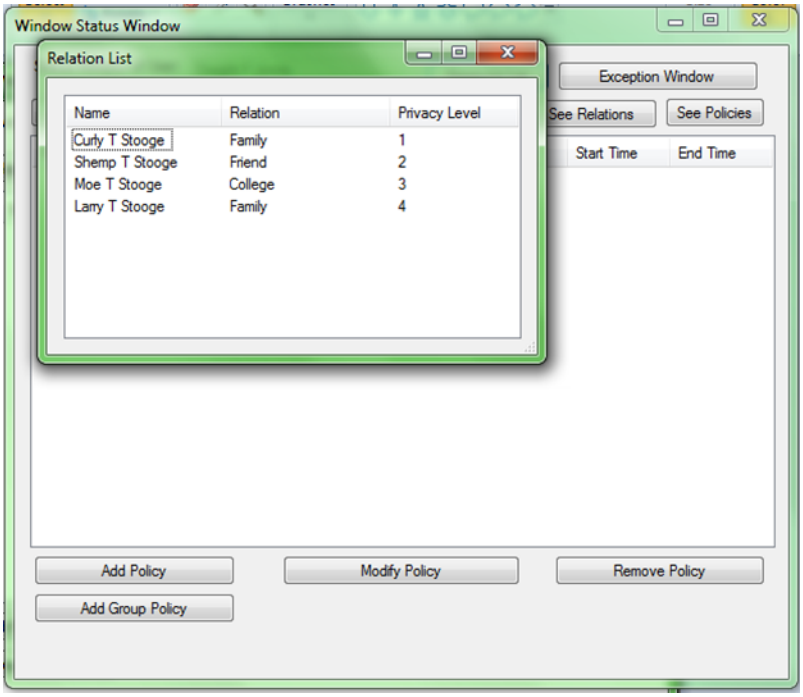

Figure 4.6 Relationship window.

4.2. EFFICIENCY

As the time for an action is based on the speed of Visual Basic, SQL, the speed of the connection and many other varying factors, a fixed measurement of time cannot be acquired. As such, estimating the time complexity in the big O notation should yield a useful measurement of time. Before talking more in-depth on the main function's time

complexity, there are a few facts that need to be established first. For one is the complexity of basic functions of SQL, insert and select.

Selecting a tuple or tuples from a table can have one of two complexities, based on whether or not the database is indexed. If a table is set up to be indexed by it primary number key, its complexity is of order O(log(n)), on the other hand if the table is not indexed its complexity becomes of order O(n). As for inserting into a database, an indexed database has an insertion complexity of O(log(n2)), because the table must be searched and the tuple to be inserted must be evaluated on where to place it. On the other hand with an unindexed table needs only to add the new tuple to the end of the table, making the complexity O(1). Secondly is the type of database to be used, which is to say whether the database is to be indexed or unindexed. While an indexed database lowers the time spent when selecting information from tables, its insertion time is much higher than that of the unindexed insertion. For this reason the model is unindexed.

Estimating the time complexity of the model is based off the main action of the system: conflict analysis, viewing other user's policies, and finally checking for your own policies. We do not need to estimate all tasks, as big O notation estimates are based on the most important aspects of the systems. The first major action the user will come across is the checking and displaying policies. This is the simplest major action of the system, as it only is doing a condition select statement. Because it is just a conditional select statement, the complexity of the statement is simply O(n). There are three tables to

check when looking for all policies, individual policies, group policies, and group privacy policy. As such because there are three checks and the base complexity becomes O(3n).

Next is finding visible users. This is a complex action, as each of the three visibility policies has its own complexity. The individual policy is a straight forward selection, making it of O(n). The group policy is of order O(nm). This is because how group policy is searched for. Let n be the number of policies in the group policies tables, and let m be the number of people in the relationship table. For each group policy, n, the entire relationship m must be checked, making the complexity, O(nm). Group privacy policy is much like group policy except it is essentially three group policies. Thus each group privacy policies, must be checked three times, one for each sub policy, making its complexity O((3n)m). Combining the complexities boors O((3m+1)nm + n), and dropping the n to simplify the order, we estimate the complexity to be O((3m+1)nm). The last major action is conflict analysis. Conflict check varies among the four policy definitions. The average number of checks for each policy is approximately three. Each checks itself is a simple search making it of order O(3n).

5. CONCLUSION

This thesis, presents a location privacy policy management system. It has three main functions. First, it helps compose location privacy policies for users who subscribe to location-based services. Second, it automatically detects policy conflict whenever there is a policy update. In this way, policy management becomes an easy task for users. Thirdly, it also provides an important feature, which is the policy recommendations. Users do not need to compose new policies for every new friend. Instead, the system will generate recommended policies based on existing privacy policies on the similar group of users. In all, the ultimate goal of our proposed system is to ease the burden on end users; when managing their privacy preferences so that they can fully enjoy the benefits of LBSs. The system prototype has been evaluated in terms of both efficiency and effectiveness.

BIBLIOGRAPHY

[1] Varshney, U. (2003) "Location management for mobile commerce applications in wireless internet environment." ACM Transactions on Internet Technology .

[2]Location Privacy Protection Act. (2001) http://www.techlawjournal.com/cong107/privacy/location/s1164is.asp.

[3] Geographic location/privacy (geopriv) group. http://www.ietf.org/html.charters/geopriv-charter.html

[4] The Economist. The end of privacy, 29th Apr 1999.

[5] The Economist. The coming backlash in privacy, 2000.

[6] Rakesh Agrawal and Ramakrishnan Srikant. (2000) "Privacy- preserving data mining." ACM Sigmod Record, May 2000 .

[7] Pierangela Samarti. (2001). "Protecting Respondents' Identities in Microdata Release." IEEE Transactions on Knowledge and Data 2001.

[8] Latany Sweeney. (2002). "Achieving K-Anonymity privacy protection Using generalization and Suppression." International Journal of Uncertainty Fuzziness and 2002.

[9] Gruteser, M. Grunwald, D. (2003) "Anonymous Usage of Location Services Through Spatial and Temporal Cloaking." systems, applications and services, 2003 .

[10] Alastair R. Beresford and Frank Stajano. (2003). "Location Privacy in Pervasive Computing." IEEE CS and IEEE Communications Society 2003.

[11] Gruteser, M.,Grunwald, D. " Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking." International Conference On Mobile Systems, Applications And Services 2003.

[12] B.Gedik and L.Liu. "Location privacy in mobile systems: A personalized anonymization model." Computing Systems, 2005. ICDCS.

[13] Sweeney, L. "Achieving K-Anonymity privacy protection Using Generalization and Suppression." International Journal of Uncertainty Fuzziness and Knowledge-based System , 2002.

[14] Sweeney, L. "K-anonymity. Amodel for protecting privacy." International Journal on Uncertainty, Fuzziness and Knowledge-based System. 2002.

[15] Gedik, B., Liu, L. “Location privacy in mobile systems: A personalized anonymization model.” Computing Systems, 2005. ICDCS 2005.

[16] Kalnis, P.,Ghinita, G., Mouratidis, K., Papadias, D. “Preserving anonymity in location based services. 2006.”

[17] Mokbel, M.F., Chow, C.Y., Aref, W.G. “The new casper: query processing for location services without compromising privacy.” (2006).

[18] Cheng, R., Zhang, Y., Bertino, E., Prabhakar., S. (2006). “ Preserving user location privacy in mobile data management infrastructures.” Privacy Enhancing, 2006 - Springer.

[19] Ghinita, G., Kalnis, P., Skiadopoulos, S. Prive: “anonymous location-based queries in distributed mobile system.” (2007).

[20] Schilit, B.N., LaMarca, A., Borriello, G., Griswold, W.G., McDonald, D., Lazowska, E., Hong, J., Iverson, V. “Challenge: ubiquitous location-aware computing and the place lab initiative.” (2003).

[21] Chow, C.Y., Mokbel, M.F., Liu, X. “A peer-to-peer spatial cloaking algorithm for anonymous location-based service.” Proceedings of the 14th annual ACM, 2006.

[22] Kido, H., Yanagisawa, Y., Satoh, T. “An anonymous communication technique using dummies for location-based services.” (2005).

[23] Snekkeness,E. “ Concepts for personal location privacy policy.” (2001).

[24] Marc Langheinrich. “Aprivacy awareness system for ubiquitous computing environments.” Ubiquitous Computing, 2002 - Springer.

[25] Sastry Duri, Marco Gruteser, Xuan Liu, Pual Moskowitz, Ronald Perez, Moninder Singh, and Jung-Mu. “Framework for security and privacy in automotive telematics.” (2002).

[26] Hengartner, U. Steenkiste, P. (2003). “Protecting Access to people Location Information.” Security in Pervasive Computing, 2004 - Springer.

[27] Langheinrich, M. “Privacy by design-principles of privacy-aware ubiquitous systems.” Ubiquitous Computing, 2001 - Springer.

[28] S Duri, M Gruteser, X Liu. “Framework for security and privacy in automotive telematics.” 2002.

[29] Ardangna, C.A., Cremonini, M., Damiani, E., de Vimercati, S.D.C., Samarati, P. “Supporting location-based conditions in access control policies.” 2006.

[30] Cranor, L.F. P3P: “Making privacy policies more useful.” IEEE Security & Privacy. (2003).

[31] Asim Smailagic, Daniel P. Siewiorek, Joshua Anhalt, David Kogan and Yang Wang. “Location Sensing and Privacy in a Context Aware.” Computing Environment IEEE Wireless Communications 2001

[32] Ginger Myles, Adrian Friday, and Nigel Davies. “Preserving Privacy in Environments with Location-Based Applications.” IEEE Pervasive Computing 2003.

[33]Gedik, B., Liu, L. “A customizable K-anonymity model for protecting location privacy.” Proceedings of the IEEE International conference on, 2005 .

[34] Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S. “Preserving user location privacy in mobile data management infrastructures.” Privacy Enhancing, 2006 - Springer.

[35] Gedik, B., Liu, L. (2007) “Protecting location privacy with personalized K-anonymity.” IEEE Transactions on Mobile Computing, 2008.

[36] Bamba, B., Liu, L., Pesti, P., Wang, T. “Supporting anonymous location queries in mobile environments with privacygrid.” (2008).

[37] Mohamed F. Mokbel. “Towards Privacy-Aware Location-Based Database Servers_ ICDEW.” (2006).

[38] Dan Lin, Elisa Bertino, and Sunil Prabhakar. “Location Privacy in Moving-Object Environments.” on Data Privacy, 2009.

[39] Bakken, D.E., Parameswaran, R., Blough, D.M., Franz,A.A., Palmer, T.J. “Data obfuscation: Anonymity and Desensitization of Usable Data Sets.” - Security & Privacy, 2004.

[40] Xiao,X., Tao, Y. “Personalized Privacy Preservation.” Proceedings of the 2006 ACM SIGMOD international, (2006) .

[41] M. Duckham and L. Kulik. “A formal model of obfuscation and negotiation for location privacy.” Pervasive Computing, 2005 - Springer .

[42] M. Duckham and L. Kulik. “ Simulation of Obfuscation and Negotiation of Location Privacy.” Spatial Information Theory, 2005 - Springer.

[43] Ardagna, C.A., Cremonini, M., Damiani, E., De Capitani di Vimercati, S. “Location Privacy Protection Through Obfuscation-Based Techniques.” 2007.

[44] Yiu, M.L., Jensen, C.S., Huang, X., Lu, H. “Space Twist: Managing the Trade-Offs Among Location Privacy.” 2008.

[45] Gandon, F.L., Sadeh, N.M.: Semantic web technologies to reconcile privacy and context awareness. 2004.

[46] Atallah, M.J., Frikken, K.B.: Privacy-Preserving Location-Dependent Query Preocessing. 2004

[47] Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.L.: Private queries in location based services: anonymizers are not necessary. 2008.

[48] Sheng Zhong, Li (Erran) Li, Yanbin Grace Liu and Yang Richard Yang Privacy-Preserving Location-based Services for Mobile Users in Wireless Networks.

[49] Danezis, G, S. Lewis, and R. Anderson, How Much is Location Privacy Worth? Hardvard University, 2005.

[50] E Kaasinen. "User Needs for Location- aware Mobile Services." 2003.

[51] J Krumm. "survey of computing location privacy, Personal and ubiquitous computing." 2008.

[52] A, Gorlach. A Heinemann. Wesley W. "Survey on Location Privacy in Pervasive Computing." 2005.

VITA

Arej A. Muhammed was born in Alabrag, Libya on May 9, 1984. She received her B.S. degree in Computer Science from Omer Almoktar University in Libya in June 2006. She worked for six months at Omer Almoktar University as teaching assistance. She joined Dr. Lin's lab group as a graduate student in August 2009 and started working towards her Master's degree in Computer Science at Missouri University of Science and Technology. She was serving as a research assistant for three years semester. She has good experience with many programing languages such as C, C++, and Visual Basic. Arej is the president of ACMW organization. In Summer 2011, she received her Master's degree in Computer Science from Missouri S&T.