Privacy Preservation in Context Aware Geosocial Networking Applications

Pramod Jagtap
Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250 USA
pramod1@umbc.edu

Anupam Joshi
Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250 USA
joshi@cs.umbc.edu

Tim Finin
Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250 USA
finin@cs.umbc.edu

Laura Zavala
Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250 USA
rzavala@umbc.edu

ABSTRACT

Recent years have seen a confluence of two major trends – the increase of mobile devices such as smart phones as the primary access point to networked information and the rise of social media platforms that connect people. Their convergence supports the emergence of a new class of context-aware geosocial networking applications. While existing systems focus mostly on location, our work centers on models for representing and reasoning about a more inclusive and higher-level notion of context, including the user's location and surroundings, the presence of other people and devices, feeds from social networking systems they use, and the inferred activities in which they are engaged. A key element of our work is the use of collaborative information sharing where devices share and integrate knowledge about their context. This introduces the need for privacy and security mechanisms. We present a framework to provide users with appropriate levels of privacy to protect the personal information their mobile devices are collecting including the inferences that can be drawn from the information. We use Semantic Web technologies to specify high-level, declarative policies that describe user's information sharing preferences. We have built a prototype system that aggregates information from a variety of sensors on the phone, online sources, and sources internal to the campus intranet, and infers the dynamic user context. We show how our policy framework can be effectively used to devise better privacy control mechanisms to control information flow between users in such dynamic mobile systems.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; H.3.4 [**Systems and Software**]: Current awareness systems—*Distributed systems*

General Terms

Security, Context-aware systems

Keywords

Privacy, social networking, mobile computing, policy

1. INTRODUCTION

Content sharing on social networking websites has dramatically increased over the last few years. Popular services such as Facebook, Twitter and MySpace allow millions of individuals to create online profiles and share personal information with a huge number of friends. The increasing availability of extended geo-location technologies such as cell tower localization on Internet services and Assisted Global Positioning System (A-GPS) on phone devices, has changed the way people interact with each other on the web. It has enriched the social networking experience with additional social dynamics that emerge from allowing users to interact relative to location and time. Location awareness is one important aspect of context-aware systems. However, context encompasses more than just the user's location, because other things of interest are also mobile and changing [16]. Other important aspects include the ambiance, resources and people nearby, and the activities in which they are engaged. The rise of online social networking systems along with recent improvements in mobile technology, smartphones, and sensor networks presents a unique opportunity for context-aware systems.

A very important but often overlooked issue in most social networking systems is that of privacy. The existing research addressing privacy issues [1], [6], [7], [10], brings out various concerns and emphasizes the need of strong privacy control

mechanisms. Furthermore, the recent emergence of context-aware geosocial networking services demands more flexible and robust access control mechanisms. These systems face similar security threats as distributed and mobile applications but privacy and trust aspects are more prominent due to the sensitive nature of contextual information. Users need to protect the personal information that their mobile devices collect through the sensors on the device, as well as the inferences that are drawn from that information. Users may simply be uncomfortable with others knowing their location, or even with their location being sensed in the first place. They can be understandably sensitive about how sensor data is captured and used, especially if it is used to reveal their location, speech , images, or video. Mobile applications such as the Audio Loop [8], which continuously record raw audio, also raise concerns and introduce issues about how (or even whether) to obtain consent to be recorded from others whose data might be captured by the user's device [9]. Such concerns could affect the adoption and use of devices that embed sensing and introduce problems into social relationships. Although there are existing approaches that can help with these problems (e.g., cryptography, privacy-preserving data mining), they are often insufficient [11].

There is a need for privacy control mechanisms that factor in the dynamic changes in the user context. Furthermore, users need to be in control of the release of their personal information at different levels of granularity, from raw sensed data to high level inferred context information. A context-aware infrastructure should provide the end user with a (logically) central place of privacy control and trust management, contrary to point solutions within different, possibly not trusted, applications [18]. Furthermore, the control mechanism ought to offer enough flexibility to allow the definition of policies for context dependent release of information. Thus, users should be able to define their privacy policies and the context-aware system should be able to protect users' information from illegal access according to the policies regardless of the application.

For instance, consider healthcare context-aware systems where sensor-enabled mobile phones can be used to collect in situ sensor data and context data such as patient's and caretakers' personal information, current location and patient's current activity. In this case a user can specify privacy policies like *"allow Dr. Nash detailed information at all time"* and *"allow access to caretaker's location only in case of emergency"*. Consider another scenario of university campus; a student user may be willing to let her teachers see where she is between 9:00am and 6:00pm on weekdays but not over the weekend. Further, she may not be willing to let her teachers know about her sleeping activity during the daytime. Additionally, a user may want to control the granularity or accuracy of the answer, depending on the current user context. For instance, she may be willing to reveal the exact room number where she to some people (e.g., coworkers), but only the city to others. Further, she may not want to disclose her location if she is at a nightclub.

Privacy control mechanisms should be flexible enough to capture contextual information about their users subject to semantically rich privacy constraints. Besides flexibility on the level of granularity of the information and the situation under which information can be shared, the incorporation of incentives can add even more richness to the policies. Consider for example, in the university campus scenario, that a particular restaurant offers discounts for groups of five or more students on a particular day. A student and a few of her acquaintances happen to be looking for lunch around that restaurant at the same time. The users might be more interested in sharing their locations under situations where they might be rewarded for doing so.

Overall, we are motivated by the need of privacy control models to control the information flow in collaborative context aware geo-social networking applications based on the user's context. None of the existing models allow users to specify the privacy preferences based on user's static and dynamic contextual information in a subtle way. Therefore, in this paper we present a policy based framework to constrain the information flow based on the user's contextual information. It can be extended and incorporated in existing social networks including location based mobile social networks. We validate our architecture in an on-campus context-aware prototype system that aggregates information from a variety of sensors on the phone, online sources, and sources internal to the campus intranet, and infers the dynamic user context. We show how our policy framework can be effectively used to devise better privacy control mechanisms to control information flow between users in such dynamic mobile systems. The remainder of the paper is organized as follows. Section 2 provides a brief overview of the related work in this area. In Section 3, we describe our representation of policies using the Semantic Web language RDF. Section 4 describes the system and various components in details, as well as the constrained information flow. We discuss our implementation in Section 5 and some evaluation experiments in Section 6. Section 7 summarizes our work with the concluding remarks and talks about the future work.

2. RELATED WORK

Context-aware systems have been studied for a long time. The focus has been mainly on the location and activity inference. The Active Badge Location system [19] used infrared technology to find the location of a user so that calls can be forwarded to phones nearby. The context-aware electronic tourist guide [5] contributed by developing location-aware tour guides which provided tourists with information depending on their location. Recently research about privacy controls in these systems has received the significant attention. AnonySense [17], a privacy-aware architecture for collaborative pervasive applications that use mobile sensing. Mobile sensor data is anonymized before its use by any of the applications. Project Aware Home [12] captures, processes and stores data (collected by sensors) about home residents and their activities. It uses access control mechanism based on Role-based Access Control (RBAC) by defining environment roles similar to subject roles of RBAC and it is used to capture security-relevant aspects of the environment in which an application executes. Context Privacy Service (CoPS) [15] describes the design and implementation of a privacy service which control how, when and to whom you could disclose a user's context information. Using the end-user survey and results of other research groups, it has identified requirements for flexible and efficient privacy service. This system is most closely related to our work. However,

it doesn't handle context-dependent privacy policies, which can be specified by users on dynamic context data. Overall, most privacy preserving works focus on location related aspects of context and deal with mechanisms to control access to such information.

3. POLICIES AND THE SEMANTIC WEB

The Semantic Web refers to both a vision and a set of technologies. The vision was first articulated by Tim Berners-Lee as an extension to the existing web in which knowledge and data could be published in a form easy for computers to understand and reason with. Doing so would support more sophisticated software systems that share knowledge, information and data on the Web just as people do by publishing text and multimedia. Under the stewardship of the W3C, a set of languages, protocols and technologies have been developed to partially realize this vision, to enable exploration and experimentation and to support the evolution of the concepts and technology.

The current set of W3C standards are based on RDF [13], a language that provides a basic capability of specifying graphs with a simple interpretation as a "semantic network" and serializing them in XML and other popular Web systems (e.g., JSON). Since it is a graph-based representation, RDF data are often reduced to a set of 'triples' where each represents an edge in the graph ('Person32 hasMother Person45') or alternatively, a binary predication (e.g., 'hasMother(Person32,Person45)'. The Web Ontology Language OWL [3] is a family of knowledge representation languages based on Description Logic [2] with a representation in RDF. OWL supports the specification and use of ontologies that consist of terms representing individuals, classes of individuals, properties, and axioms that assert constraints over them. The axioms can be realized as simple assertions (e.g., 'Woman is a sub-class of Person', 'hasMother is a property from Person to Woman', 'Woman and Man are disjoint') and also as simple rules.

The use of OWL to define policies has several important advantages that become critical in distributed environments involving coordination across multiple organizations. First, most policy languages define constraints over classes of targets, objects, actions and other constraints (e.g., time or location). A substantial part of the development of a policy is often devoted to the precise specification of these classes, e.g., the definition of what counts as a 'student' or a 'entertainment activity'. This is especially important if the policy is shared between multiple organizations that must adhere to or enforce the policy even though they have their own native schemas or data models for the domain in question. Second, OWL is based on description logic, a well understood subset of logic for which powerful and efficient reasoning systems are available. By constraining our use of OWL to the right subset, we can exploit existing OWL reasoners. A third advantage is that OWL's grounding in logic facilitates the translation of policies expressed in OWL to other formalisms, either for analysis or for execution. Finally, OWL is designed of and for the Web, making sharing policies and the ontologies they use both natural and easy.

4. SYSTEM ARCHITECTURE

The proposed system architecture is shown in the Figure 1. The major components of this system are client devices, server side modules and the Internet services that provide social media. The client devices are location aware smartphones. Today's smartphones are programmable and come with a large set of cheap powerful embedded sensors, such as a camera, GPS, accelerometer, digital compass, gyroscope, microphone, and many more. These sensors are enabling the emergence of personal, group and community scale sensing applications. These client devices as well as the server side modules contain a user profiles repository, a privacy control module and content preferences. The server side also contains a content aggregator, a learn and share module and a privacy control module. The content aggregator combines social media like event updates, photos, and videos from Internet services like YouTube, Flickr, Facebook or university information portals. The learn and share module infers a user's dynamic context using sensor data collected by a variety of sensors on the phone, the information from the content aggregator and online sources such as user's calendar. The inferred context is shared with corresponding client device so that the device along with server can handle further context sharing queries from other clients. The requester queries are passed through the privacy control module to constrain the information flow and hence to protect the user privacy. The privacy control module provides the access control mechanisms and aids in controlling the information flow within system. On the client device, it enables privacy sensitive and resource sensitive reasoning over sensed data along with privacy enforcement between peer devices sharing contextual information. The interaction between various components of our system can be described as follows:

- The user of the system has a client device to collect the sensor data periodically. This data is passed to the learn and share module on the server as allowed by the privacy control module on the client device. The privacy control module decides which specific sensor data can be shared with the server based on user-specified privacy policies.
- The learn and share module infers the user's context using sensor data and information from the content aggregator and other online sources. This context consists of current location, activity and additional surrounding information like nearby people. The inferred knowledge is passed to the corresponding client device so that it can handle context access queries from other clients.
- These access requests are passed through the privacy control module which in turn decides whether to allow or deny the access. If the requester is granted the access then it determines a set of information to be shared by performing reasoning over the context information and user's privacy preferences. These requests can be made by one client device to another or from a client device to the server.
- Figure 1 shows the three different ways in which information can be shared in our system, namely: (i) context information sharing between the client devices, (ii) sensor data sharing between a client device and the server, and (iii) context information sharing between a

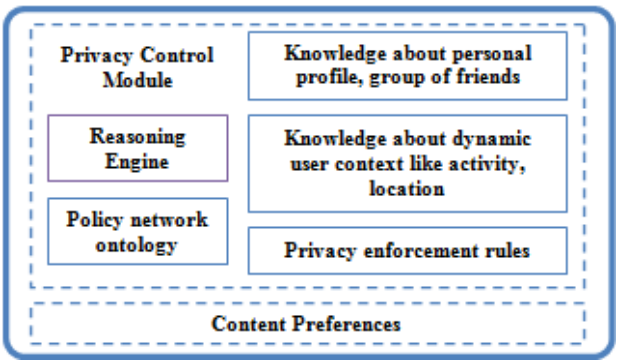

Figure 2: Our privacy control module is supported by a common policy ontology and an OWL reasoning engine. It enforces a users' information sharing policies using both static information about the user as well as dynamic information observed and inferred from her context.

client device and the server. The information sharing is controlled by the privacy control module in order to preserve user privacy.

We will focus our discussion on our privacy mechanisms and the relevant system components which have most direct influence on the information flow in the system.

4.1 Privacy Related Components

The privacy control module aims to protect the owner's context information by performing reasoning over her context. It deals with the resource to be protected, the owner of a resource and the requester who wants to access it. It has access to owner's profile information and the group information along with specified privacy policies. By performing the reasoning over the context data it enforces set of privacy policies specified by the owner. As shown in (Figure 2), it consists of, (i) a set of ontologies for describing policies and access requests, (ii) a reasoning engine that accepts requests and performs the reasoning, (iii) the knowledge about the owner and (iv) the privacy preferences.

4.1.1 Context ontology

The context-aware systems raise the need of models for representing and reasoning about a more inclusive and higher-level notion of the context. Our context model ontology captures the user location and surroundings, the presence of other people and devices, and the inferred activities in which they are engaged. We adopt description logics (DL), specifically OWL (Web Ontology Language), and associated inferencing mechanisms to develop a model of context and policies. In the ontology model, the actions are in general lower level tasks and have no associated role. The activities are introduced as means to abstract multiple actions and further, to associate roles to the sets of actions. Places can be defined in terms of the activities that occur there. Ambiance includes concepts describing the environment of the principal (e.g., noise level, ambiance light, and temperature). The context ontology as shown in Figure 3, captures the semantic notion of context in a mobile context-aware system. Using the ontology, each device contains a declarative knowledge

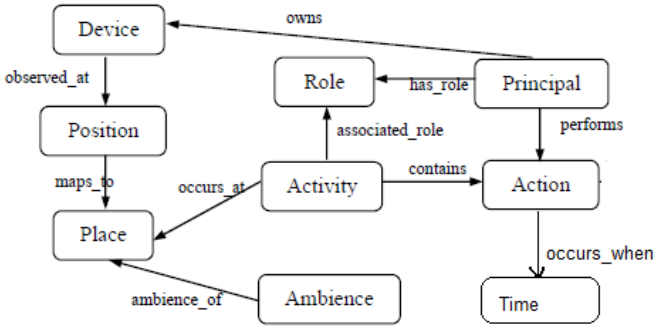

Figure 3: The Context Ontology models the key concepts of context.

base with semantically rich information about user's information, activities, inferences, and further contextual information. The knowledge base aligns with the context ontology which defines the key context concepts used for making access control decisions.

4.1.2 Reasoning Architecture

The resoning engine handles the requester queries and performs reasoning for access control decisions. Our system uses the Jena Semantic Web framework[4] for performing the reasoning over context data. Jena inference system allows the support of various inference engines or reasoners. These reasoners are used to infer additional facts from the existing knowledge base coupled with ontology and rules. In particular, Jena uses the generic rule reasoner which is included in Jena2 as a general purpose rule-based reasoner. It is used to implement both the RDFS and OWL reasoners. It needs at least a rule set to define its behavior. ItŠs instance with a ruleset can be used like any of the other reasoners - that is it can be bound to a data model and used to answer queries to the resulting inference model. in our system, the reasoning engine uses the context ontology, userŠs context information and group information along with the user-specified privacy rules to generate an inference model. This inference model is used for responding to the requester queries. This process is shown in the Figure 4 and explained in the following steps:

- We create the instance of OWL reasoner specialized for context ontology and then apply that to the userŠs profile and group information to generate an inference model. This inference model consists of additional statements inferred from static knowledge and ontology. As the user information and ontology arenŠt changed often, it is quite safe to save the model on external storage and reload it for subsequent queries rather than generating it each time. The save and reload is an optional step used for optimizing performance on mobile devices.
- The user's contextual information is added to the inference model to generate a new inference model.
- In this step, the user-specified privacy rules are executed with the previously generated inference model to generate a new inference model having requester

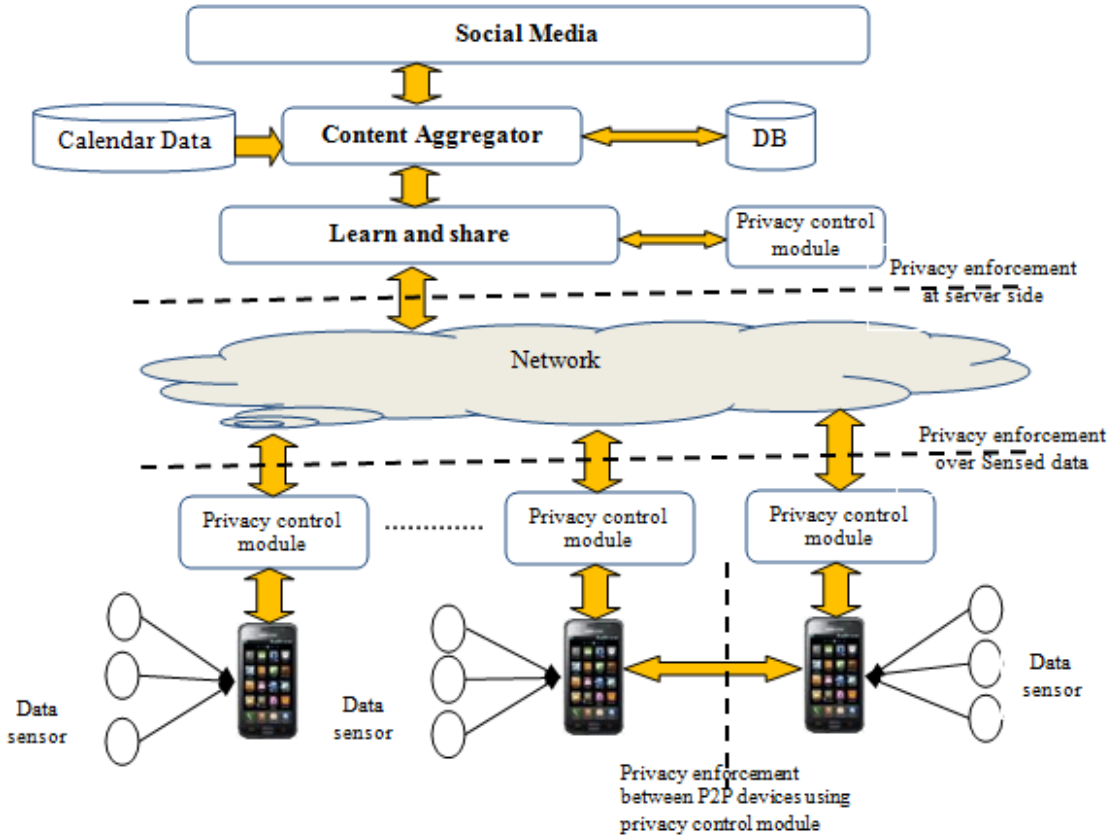

Figure 1: The Architectural view of the system

access levels. The system will use the new model to decide what can be shared with requester and respond accordingly.

4.1.3 *Knowledge about the user*

A user can create her personal profile and put in information like name, email address, hobbies and interests and can manage different groups of her friends. Apart from that, the system has dynamic knowledge information about user including current activity and her recent location. Our context ontology defines the entities required to represent a user information in addition to the FOAF vocabulary. This knowledge is specified using N3 in our system. The context sensitive information such as a user's current location can be edited by the user and is accepted by the system with consent. All the attributes in a user's personal profile as well as data sensed by mobile devices are considered as resources to be protected. Every protected resource has a governing policy that is updated whenever the owner of the resource changes the privacy preferences. The privacy control module in the system acts as a guard. When a client requests access to a protected resource, it checks if there is a policy associated with the resource. The sample user information is shown in Table 1. Here, ex is the namespace of user information file and foaf represents the FOAF vocabulary. It states that Harry is a person belonging to the Family group.

4.1.4 *Privacy preferences*

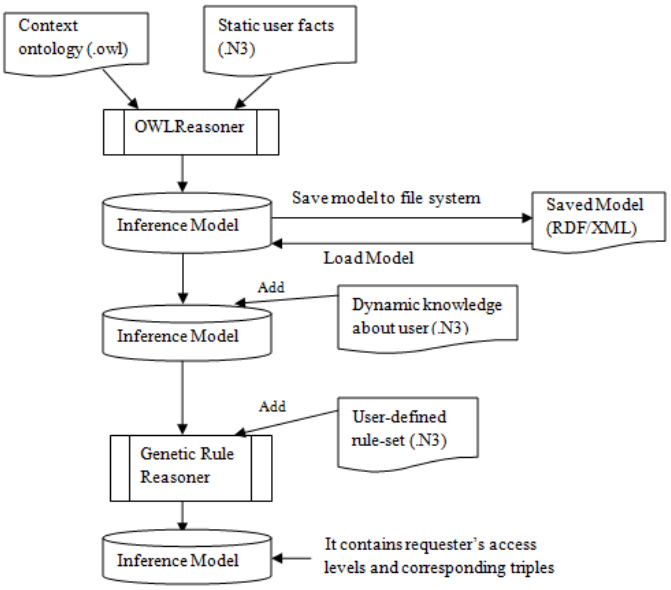

Figure 4: The reasoning architecture.

Privacy preferences are access control rules that describe how a user wants to share which information, with whom, and under what conditions. A user can disclose information

Table 1: Sample user information

```
ex:Harry a foaf:Person ;
    foaf:name "Harry" ;
    ex:memberOf ex:GroupFamily .
ex:GroupFamily a foaf:Group ;
    foaf:name "Family" .
```

with different accuracy levels; for instance, she may tell the exact building on the university campus she is in to her close friends, but just the county or town she is in to others. A user may decide not to disclose her location to advertisers. A user can manage different networks of friends, and assign variety of group level privacy preferences accordingly. For instance, a user can create a group of family members, a group of colleagues, or a group of teachers, and may define distinct privacy settings for each of them. Conditions can be defined based on attributes like a user's current location, current activity or any other dynamic attribute. A user can also define rules for advertisers. All the privacy preferences are represented as N3 rules in the system.

4.2 Privacy Preservation

The user's personal information can be shared between a client device and the server side application or between two client devices. To constrain the information flow, privacy enforcement can be done on (i) client devices over sensed data, (ii) on peer client devices and (iii) at server side for contextual information.

4.2.1 Privacy enforcement between peer client devices

The learn and share module from server side shares the owner's contextual information with corresponding client device. The client device further keeps track of the context and responds to queries made by other peer devices. Table 2 shows the sample contextual information for user "Alice". This contextual information needs to be protected and should be shared only with requesters having sufficient privileges. The user can provide detailed privacy policies specifying what context information can be shared with whom, when, and under what conditions. If users are reluctant to provide any specific policies then they can opt for either default models of the system viz. (i) Optimistic Model - where the system can provide response to any query with all possible relevant information associated with a user's activity such as associated place, location and the timing details, or (ii) Pessimistic Model - where the system can refrain from revealing activity associated information. Apart from these default system settings the user can define her privacy rules with various degrees of accuracy levels. She can also use the system to obfuscate certain pieces of information to protect the context information. This way our system can protect the user's privacy by varying accuracy levels of activities, associated locations and timestamps.

Whenever any participant in the systems tries to access any protected resource (activity, place, location or any additional information) the query is sent to the privacy control module.

Table 2: Contextual information represented in N3. It consists of activity, associated place, location, time and nearby users.

```
ex:Alice a foaf:Person ;
    foaf:name "Alice" ;
    platys:has_role platys:Student .
platys:Sleeping a platys:Activity ;
platys:is_performed_by ex:Alice ;
    platys:has_participant ex:Alice, ex:John ;
    platys:occurs_at platys:Class_LH1 ;
    platys:occurs_when "2010-11-19T14:12:42".
platys:Class_LH1 a platys:Place ;
    platys:has_location "39.253525, -76.710706".
```

This module fetches the user knowledge, dynamic knowledge and user-specified privacy preferences to evaluate the query. As a result it will decide whether participant is allowed to access to protected resource or not. In former case, it might obfuscate certain pieces of the information as per user-specified privacy policies to protect user privacy. Our system uses Jena [4] on Android device [14] to perform reasoning and constraining sensed data flow according to user-defined privacy policies.

For example, suppose a user has privacy policies like the following: (i) *Share detailed contextual information with family members all the time.* (ii) *Share my activity with friends all the time except when I am attending a lecture.* (iii) *Do not share my sleeping activity with Teachers on weekdays from 9am-9pm.* These policies are represented as Jena rules in Table 3, Table 4 and Table 5 respectively.

At any given time, if the request comes from a requester "Ron" who is a family member of the user then he should be able to access the user's detailed contextual information. If the request comes from a requester "Bob" who is a member of the friend group and the user's current activity is "Sleeping" then the requester is allowed to access a user's activity information excluding the associated place and location. Figure 5 shows the access level for requester "Ron" and Figure 6 shows allowed access for user "Bob" after performing reasoning on android device using user information, dynamic knowledge and privacy policies mentioned in Table 3, Table 4 and Table 5.

4.2.2 Privacy enforcement over the sensed data

The sensor date collected by client devices is sent to the server for inferring a user's dynamic context. As users can be sensitive about how sensor data is captured and used, it is best to let them control how their sensor information is released. It can be done by providing users with an option to specify privacy policies to protect the sensed data. Before data is collected from sensors in continuous sensing or whenever there is a request for sensed data, the privacy control module evaluates the user-defined privacy policies and decides which sensor data can be collected. Only allowed sensors' data is collected and sent to the server for further context inferring. For instance, user can have policy like

Table 3: Policy to share detailed contextual information with family members

```
[AllowFamilyRule:
    (?requester ex:memberOf ?groupFamily)
    (?groupFamily foaf:name “Family”)
->
    (?requester ex:canAccessActivity “True”)
    (?requester ex:canAccessActivityPlace “True”)
    (?requester ex:canAccessActivityTime “True”)
    (?requester ex:canAccessPlaceLocation “True”)
]
```

Table 4: Policy to share activity information with friends all the time except when a user is attending lecture

```
[ShareActivityWithFriendsRule:
    (?requester ex:memberOf ?groupFriends)
    (?groupFriends foaf:name “Friends”)
    (?someActivity platys:is_performed_by ex:Alice)
    notEqual(?someActivity, platys:Listening_To_Lecture)
->
    (?requester ex:canAccessActivity “True”)
]
```

Table 5: Policy to not share sleeping activity with Teachers on weekdays from 9am - 9pm

```
[ShareActivityWithTeachersRule:
    (?requester ex:memberOf ?groupTeachers)
    (?groupTeachers foaf:name “Teachers”)
    (?requester ex:requestTime ?localTime)
    (?localTime time:dayOfWeek ?day)
    ge(?day, 1) le(?day, 6) (?localTime time:hour ?hour)
    ge(?hour, 9) le(?hour, 21)
    (?someActivity platys:is_performed_by ex:someUser)
    equal(?someActivity, platys:Sleeping)
->
    (?requester ex:canAccessActivity “False”)
]
```

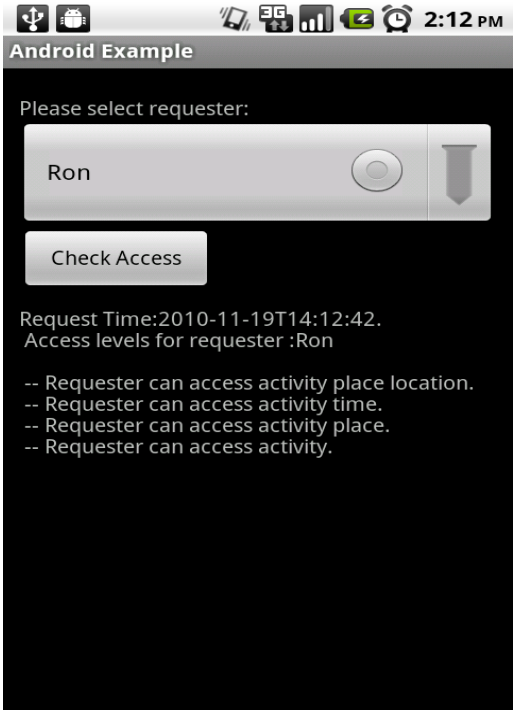

Figure 5: Android device screen with reasoning results. It has access levels for requester “Ron” who belongs to family member group.

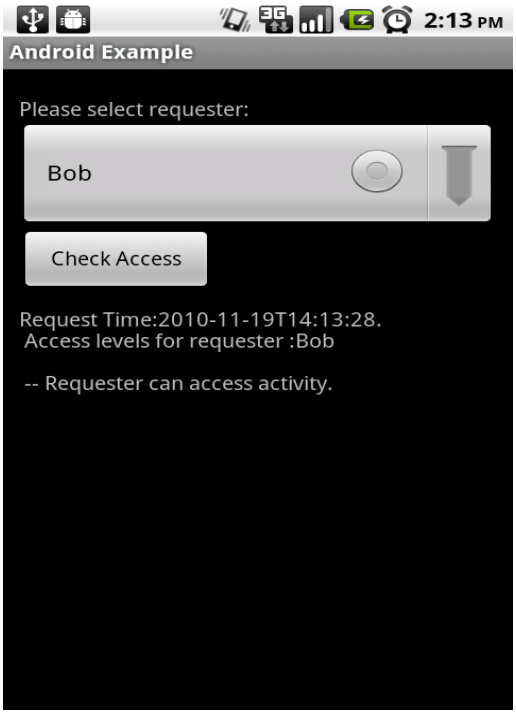

Figure 6: Android device screen with reasoning results. It has access levels for requester “Bob” who belongs to friend group.

“share GPS co-ordinates on weekdays from 9am-5pm only if

Table 6: Policy to share GPS coordinates. It states that GPS data can be shared on weekdays from 9am-5pm only if user is in office.

```
[ShareGPSRule:
    (?requester ex:requestTime ?localTime)
    (?user ex:systemUser ?true)
    (?localTime time:dayOfWeek ?day)
    ge(?day, 1) le(?day, 6)
    (?localTime time:hour ?hour)
    ge(?hour, 9) le(?hour, 17)
    (?user ex:Latitude ?latitude)
    (?user ex:longitude ?longitude)
    Equal(?latitude, ?officeLat)
    Equal(?longitude, ?officeLong)
->
    (?requester ex:canAccessGPSCoordinates "True")
    (?requester ex:canAccessActivityPlace "True")
    (?requester ex:canAccessActivityTime "True")
    (?requester ex:canAccessPlaceLocation "True")
]
```

Table 7: Policy to share accelerometer readings, WiFi AP ids and recorded audio. It states that accelerometer and WiFi information will be shared on weekdays only but audio information won't be shared.

```
[ShareAccelerometerRule:
    (?requester ex:requestTime ?localTime)
    (?localTime time:dayOfWeek ?day)
     ge(?day, 1) le(?day,6)
->
    (?requester ex:canAccessAccelerometerReadings "True")
    (?requester ex:canAccessWiFiIds "True")
    (?requester ex:canAccessAudioData "False")
]
```

he is in office". Table 6 shows it's corresponding Jena rule.

In another case, a user can have policy like *"Do not allow access to recorded audio but allow access to accelerometer and WiFi AP ids on weekdays".* Table 7 shows corresponding Jena rule syntax.

4.2.3 Privacy enforcement at the server side

At the server side learn and share module, infers the user's dynamic context such as current activity, associated place and location and nearby people. This contextual information needs to be protected and should only be shared with requesters with sufficient privileges. The server has information about all the system users whereas a client device has information about it's owner. Due to this, the server can handle requests for all the users whereas the client device can handle requests about it's owner only. The main distinction between the access requests made by a client device to a peer device and to a server is that the latter request contains a specific userId. This userId is used to retrieve specific user's information. Consider a privacy policy as shown in the table 8, which states *"allow location access to teachers on weekdays only between 9am and 6pm".* The system uses the userId to retrieve the related information and then checks whether the requester is a member of the group by verifying the requester's userid. The example explained above involves representation of a user's personal resources such as list of friends, groups information, contextual attributes like current location and current activity.

Table 8: Policy to share location with teachers on weekdays only between 9am and 6pm

```
[ShareActivityWithTeachersRule:
    (?requester ex:memberOf ?groupTeachers)
    (?groupTeachers foaf:name "Teachers")
    (?requester ex:requestTime ?localTime)
    (?localTime time:dayOfWeek ?day)
    ge(?day, 1) le(?day, 6)
    (?localTime time:hour ?hour)
    ge(?hour, 9) le(?hour, 18)
    (?user ex:systemUser ?true)
    Equal(?user, ?userId)
->
    (?requester ex:canAccessPlaceLocation "True")
]
```

5. IMPLEMENTATION

We have used location-aware devices such as iPhone or Google Android phone as client devices in our prototype implementation. Our mobile application collects the sensor data and sends it to the server for processing. The server side module has provision to collect data from various online sources such as Google Calendar or social networking sites such as Facebook. This module can collect user profile information and find networks of their friends. Using this information with sensor data this module can infer contextual information of user. For requesting the context of a user, the system provides a Web user interface at client device that consists of Google maps mashed up to plot user and her friends. The requester can select a friend from her friend list and query the location or activity. The query is processed by the policy framework and it's result is shown to the requester with valid accuracy level. In the implementation, we have used contextual information as the resource that changes dynamically for the user, and have provided mechanisms to specify more expressive policies to control the sharing of contextual information. The users can create policies by using Policy Editor interface as explained below.

5.1 Specifying a privacy policy

The users can use the Web interface from client device to specify and edit privacy preferences. They can specify access control rule as - 'who' by selecting friends or groups of friends, 'what' by selecting resources such as location or activity, 'conditions' by selecting allowed days of the week or specifying the allowed time range during day or by specifying

Figure 7: Privacy editor for client devices. It allows users to specify and configure various privacy rules.

region on the map as sensitive. Users can also specify allowable type of activity like sleeping, eating, working, chilling. Figure 7 describes the sample privacy rule editor for client devices. The various example policies discussed in section 4 has been tried out in the implementation. The policies are created and stored in N3 format on both server and client sides in persistent memory and reloaded when required by reasoning engine. The current implementation does not provide user interface to generate policy required for the explain justification of the policies.

Our primary goal for the prototype was to use Semantic Web based policy framework to demonstrate strong access control over the static and transient user information in a collaborative context-aware geosocial networking system.

6. SYSTEM EVALUATION AND RESULTS

We have built an android application to collect sensor data and to perform reasoning over sensed data and contextual information using Jena. It fetches user information, user group information, dynamic knowledge about user and user-defined privacy policies from device memory. Whenever this application is started, it loads the user profile and group information model. Before collecting the sensed data or responding to peer's request it checks whether user information model is loaded or not. After successfully loading user information model it reads user-defined privacy policies, contextual information and performs reasoning to provide access levels. We used Nexus one devices having Google Android 2.2.1 operating system to run this application and collect data.

The goals of evaluation were (i) to see if the system satisfies a basic criteria by allowing access from privileged user and restricting illegal user, (ii) to test whether the actual computing time of reasoning over mobile devices is acceptable and (iii) to determine how it scales with different size of user information like number of users in group list. The system behaved as expected by allowing information access to privileged users and denying access to illegal users as per user-defined privacy rules. Here, we define a priviledge user as a requester who is allowed to access user's context as per user-specified privacy rules whereas other's are modelled as illegal users. To evaluate scalability of the system, we varied the number of users in group list and noted the time taken (response time) by the system to provide access levels for the requester when (i) user information model wasn't loaded in memory and (ii) user information model was already loaded. Table 9 shows the results of the evaluation where obtained values are average of several computations. The is evident that, the system takes a few seconds to load large user information model and process the query but once model is loaded, further requests are processed without any significant overhead. It shows that reasoning on mobile devices can be done without any scalability issues and it can be efficiently used to enforce privacy over sensed and contextual data. Figure 8 shows the linear growth of response time (in milliseconds) against number of users in the group list.

Table 9: Response time for different number of users

Numbers of users	Initial response time		Response time for subsequent requests(ms)
	Response time(ms)	Standard deviation	
10	1701	12.72	121
50	2681	56	144
100	2958	93.97	136
250	3404	121.36	162
500	4435	98.66	228
1000	6556	104.18	310

7. CONCLUSION AND FUTURE WORK

Our mobile devices are becoming the dominant way we communicate with people, access information, and consume services. As they become more intelligent, they can and will model our interests, activities and behavior in order to understand our current context and using it, better serve our needs. When appropriate, aspects of this leaned context may be shared with other devices in order to collaborate and provide enhanced service. This development introduces a strong need to allow users greater control of what information is shared with who and with what level of detail.

We described a policy based framework to control information flow in collaborative context aware geo-social networking application. It allows users to specify a rich suite of privacy preferences that consider the static and dynamic knowledge about user, along with generalization rules to regulate

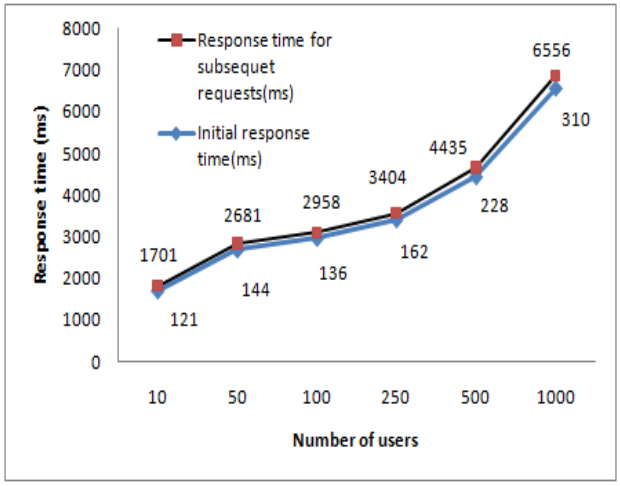

Figure 8: Response time (in milliseconds) for different number of users' in owners group list. Initial response time indicates that time taken to evaluate a query by loading user information model and performing reasoning whereas Response time for subsequent requests indicates only time taken to evaluate query by performing reasoning on already loaded model.

the accuracy of results. Protected resources can be activities, location information, or media such as photos, videos posted by participants of the social network. We showed some example policies that state of the art systems do not support. Our privacy mechanisms constitute a baseline that can be extended and incorporated by any of the existing social networks including location based mobile social networks. We plan to extend the prototype implementation to address the engineering challenge of scalability. We plan to carry out user studies to evaluate the utility of the proposed privacy control mechanisms. We also plan to address the issues of incorporating incentives to allow for even more flexibility in the definition of policies for context-dependent release of information.

8. ACKNOWLEDGMENTS

The research described in this paper was supported by the National Science Foundation (award 0910838) and the Air Force Office of Scientific Research (MURI Grant FA9550-08-0265)

9. REFERENCES

[1] A. Acquisti and R. Gross. Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. In *Proceedings of 6th Workshop on Privacy Enhancing Technologies*, 2006.

[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, New York, NY, USA, 2003.

[3] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. Mcguinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language Reference. Technical report, W3C.

[4] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: implementing the semantic web recommendations. pages 74–83, New York, NY, USA, 2004. ACM.

[5] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou. Developing a context-aware electronic tourist guide: some issues and experiences. In *CHI*, pages 17–24, 2000.

[6] C. Dwyer, S. R. Hiltz, and K. Passerini. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In *Proceedings of the Thirteenth Americas Conference on Information Systems (AMCIS)*, 2007.

[7] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, WPES '05, pages 71–80, New York, NY, USA, 2005. ACM.

[8] G. R. Hayes, S. N. Patel, K. N. Truong, G. Iachello, J. A. Kientz, R. Farmer, and G. D. Abowd. The personal audio loop: Designing a ubiquitous audio-based memory aid. In *In Proceedings of Mobile HCI*, pages 168–179. Springer Verlag, 2004.

[9] G. Iachello, K. N. Truong, G. D. Abowd, G. R. Hayes, and M. Stevens. Prototyping and sampling experience to evaluate ubiquitous computing privacy in the real world. In *Proceedings of the SIGCHI conference on Human Factors in*, page 1009. Press, 2006.

[10] H. Jones and J. Soltren. Facebook: Threats to privacy. ethics and the law on the electronic frontier course, 2005.

[11] A. Kapadia, D. Kotz, and N. Triandopoulos. Opportunistic sensing: security challenges for the new paradigm. pages 127–136, Piscataway, NJ, USA, 2009.

[12] C. Kidd, R. Orr, G. Abowd, C. Atkeson, I. Essa, B. MacIntyre, E. Mynatt, T. Starner, and W. Newstetter. The Aware Home: A Living Laboratory for Ubiquitous Computing Research. volume 1670, pages 191–198. 1999.

[13] O. Lassila and R. Swick. Resource description framework model and syntax specification, 1998.

[14] Lorecarra. Androjena : Jena android porting, 2009.

[15] V. Sacramento, M. Endler, and F. N. Nascimento. A privacy service for context-aware mobile computing. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 182–193, 2005.

[16] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *In Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85–90, 1994.

[17] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos. AnonySense: A system for anonymous opportunistic sensing. *Journal of Pervasive and Mobile Computing*, 2010.

[18] M. e. a. van Sinderen. Supporting context-aware mobile applications: an infrastructure approach, 2006.

[19] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10:91–102, January 1992.