Protecting Access to People Location Information

Urs Hengartner[1] and Peter Steenkiste[1,2]

[1] Computer Science Department
[2] Department of Electrical and Computer Engineering
Carnegie Mellon University
{uhengart,prs}@cs.cmu.edu

Abstract. Ubiquitous computing provides new types of information for which access needs to be controlled. For instance, a person's current location is a sensitive piece of information, and only authorized entities should be able to learn it. We present several challenges that arise for the specification and implementation of policies controlling access to location information. For example, there can be multiple sources of location information, policies need to be flexible, conflicts between policies might occur, and privacy issues need to be taken into account. Different environments handle these challenges in a different way. We discuss the challenges in the context of a hospital and a university environment. We show how our design of an access control mechanism for a system providing people location information addresses the challenges. Our mechanism can be deployed in different environments. We demonstrate feasibility of our design with an example implementation based on digital certificates.

1 Introduction

Ubiquitous computing environments, such as the ones examined in CMU's Aura project [1], rely on the availability of people location information to provide location-specific services. However, location is a sensitive piece of information and releasing it to random entities might pose security and privacy risks. For example, to limit the risk of being robbed, individuals wish to keep their location secret when walking home during the night. Therefore, only authorized entities should have access to people location information.

Whereas location information has received increased attention, its access control requirements have not been studied thoroughly. Location information is inherently different from information such as files stored in a file system, whose access control requirements have been studied widely. Location information is different since there is no single point at which access needs to be controlled. Instead, a variety of sources (e.g., a personal calendar or a GPS device) can provide location information. In addition, different types of queries can provide the same information (see Section 3.4). Therefore, a system providing location information has to perform access control in a distributed way, considering different services and interactions between queries.

The contribution of our work is threefold. First, we discuss challenges that arise when specifying access control policies for location information in different environments. Second, we provide the design of an access control mechanism that is flexible enough to be deployed in different environments, having multiple sources of location information. Third, we present an implementation of the proposed mechanism, which is based on digital certificates.

The outline of the rest of this paper is as follows: We introduce the architecture of a people location system in Section 2. In Section 3, we discuss several challenges that arise when specifying location policies. We explain how we deal with multiple sources of location information in Section 4. In Section 5, we present the design of our access control mechanism. We discuss our prototype implementation in Section 6. We comment on related work in Section 7 and on our conclusions and future work in Section 8.

2 People Location System

In this section, we introduce the architecture of a people location system that exploits different sources of location information.

We assume that the location system has a hierarchical structure. Figure 1 shows an example of such a system, as it could be deployed in CMU's School of Computer Science (SCS). The nodes in the graph are either services or devices, the arrows denote which service contacts which other service or device. The *location system* is a composition of multiple *location services*. Each location service either exploits a particular technology for gathering location information or processes location information received from other location services. Location information flows in the reverse direction of a request (not shown in the figure). A location service can be implemented either on a single host or on multiple hosts to improve scalability and robustness.

There are two groups of location services. The first group consists of services that are aware of the location of people. The second group includes services that are aware of the location of devices. These services locate a user indirectly by locating the device(s) the user is carrying with her. The People Locator service, the Calendar service, and the Device Locator service belong to the first group. The People Locator service aggregates information received from other services. The Calendar service looks at people's appointments to determine their current location. The Device Locator service maps a query for a person to potentially several queries for her devices and contacts corresponding services in the second group. In our example, this group of services consists of the Wavelan service and the GPS service. The Wavelan service keeps track of the location of wireless devices by identifying their base station. The GPS service retrieves the location from GPS-enhanced mobile phones. We believe that our location system can easily incorporate other location services (e.g., Microsoft's Radar [2] or MIT's Cricket [3]).

A basic assumption in our work is that different organizations may administer the various services. In our example, SCS's computing facilities control the

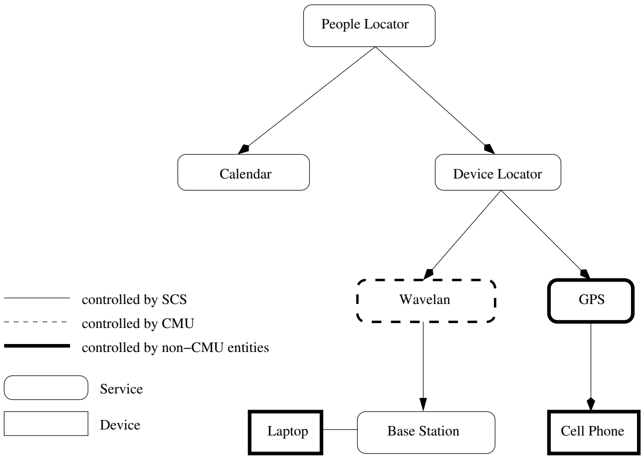

Fig. 1. Example location system. Clients query the People Locator service for the location of a person. This service forwards a query to the Calendar service and to the Device Locator service. The Device Locator service locates a person by locating her devices; namely, it queries the Wavelan service for her laptop and the GPS service for her cell phone.

Calendar service, CMU's computing facilities administer the Wavelan service, and a phone company runs the GPS service.

3 Location Policies

Location queries can originate both from people and from services. In the rest of this paper, we are going to call an entity that issues a query a *location seeker*. A query can ask either for the location of a user (*user query*) or for the people in or at a geographical location, such as a room in a building (*room query*). Based on these two basic queries, it becomes possible to build more sophisticated queries or services that provide location-specific information.

To prevent location information from leaking to unauthorized entities, we employ location policies. An entity can access location information about a person or about the people in a room only if permitted by that person's and that room's location policy, respectively. In this section, we examine location policies and present requirements that need to be provided by the access control mechanism of a people location system.

3.1 User and Room Policies

Corresponding to the two types of queries, there are two types of location policies: user policies and room polices. A user policy states who is allowed to get location information about a user. For example, "Bob is allowed to find out about Alice's location". Similarly, a room policy specifies who is permitted to find out about the people currently in a room. For example, "Bob is allowed to find out about the people in Alice's office".

In addition, both user and room policies should be able to limit information flow in other ways. Namely, we believe that at least the following properties should be controllable:

Granularity. A policy can restrict the granularity of the returned location information. For example, a user policy can state that the building in which a queried user is staying is returned instead of the actual room (e.g., "CMU Wean Hall" vs. "CMU Wean Hall 8220"). A room policy can require that the number of people in a room is returned instead of the identity of the people in the room (e.g., "two people" vs. "Alice and Bob").

Locations/users. User policies can contain a set of locations (e.g., buildings or rooms). The location system will return location information only if the queried user is at one of the listed locations. For example, "Bob is allowed to find out about Alice's location only if she is in her office". Similarly, room policies can include a set of users. The answer to a room query will include only users listed in the policy (provided they are in the room).

Time intervals. Location policies can limit time intervals during which access should be granted. For example, access can be restricted to working hours.

Earlier work (e.g., by Spreitzer and Theimer [4]) lets users place boundaries on room policies. Namely, users can specify whether they want to be included in results to room queries. While this approach is appropriate for some scenarios, it is not for others. We argue that in most cases, the owner of a room should always be able to find out who is in her room, regardless of the user policies of the users in her room.

3.2 User vs. Institutional Policies

Depending on the environment, different entities specify location policies. For some environments, a central authority defines policies, whereas for others, users set them. In addition, some environments might give both users and a central authority the option to specify policies.

In general, governments and companies probably do not want the location of their employees or the people in their buildings to be known to outsiders, whereas this information can be delivered to (some) entities within the organization. In such cases, a central authority would establish the location policies such that no information is leaked. For other environments, such as a university or a shopping mall, the institution behind the environment cares less about where an individual is or who is in a room. For these cases, it should be up to an individual to specify her user policy. We examine some example environments in more detail in Section 3.6.

In the rest of the paper, we are going to call the entity that specifies location policies *policy maker*. A location system should be flexible enough to support different policy makers, depending on the environment.

3.3 Transitivity of Access Rights

If Bob is granted access to Alice's location information, should he be allowed to forward this access right to Carol? If Ed is given the right to find out about the people in his office, should he be allowed to grant this privilege also to Fred? In short, should access rights to location information be transitive?

There is no simple answer to this question. Again the answer depends on the environment. The location system should let policy makers explicitly state whether they want access rights to be transitive. Note that even though a user might not be allowed to forward his access rights to other users, he could still issue queries on their behalf. The only way to deal with this problem is to take away the access right from this user.

3.4 Conflicting Policies

User and room policies can conflict. For example, assume that Alice does not allow Bob to locate her, but Carol allows Bob to locate people in her office. If Alice is in Carol's office, should the location system tell Bob about it? There are multiple ways for dealing with this issue:

- The system ignores the room policy when answering a user query. Similarly, it ignores the user policies for a room query. In our example, Bob would thus see Alice if he asks for the people in Carol's office, but he would not be allowed to ask a user query for Alice.
- The system looks at both policies for any request and returns information only if it is approved by both of them. Bob would thus never see Alice being in Carol's office.
- The user and room policies are established in a synchronized fashion so that no conflicts arise. For example, Leonhardt and Magee [5] suggest authorizing user/room pairs. Alice and Carol's location policies would thus have to be rewritten.

The approach that fits best depends again on the environment in which the location system is deployed.

3.5 Privacy Issues

Location policies can contain personal data that users might want to keep private. For example, a user might grant access rights to his friends and thus define a set of friends. The decision about who is (not) in this set can be delicate, and the user wants to keep the identities of people in the set secret. A location system should allow users to keep their policies secret.

3.6 Example Environments

In this section, we discuss how location policies are specified and applied in two different environments; a hospital and a university environment.

Hospital. Medical data, such as patient information, is typically protected based on a multilateral security model, which protects information flow between compartments. For example, only doctors taking care of a patient have access to her medical data, but not every doctor in the hospital. A similar model is required for location information. Only the doctors of a patient should be able to locate her. In addition, a patient should be able to allow other people (e.g., her husband) to locate her. To fulfill these requirements, the hospital has a central authority establish policies. It can give the patient the right to include additional people in her user policy.

Room policies should be established by the central authority to protect the patients' privacy. User and room policies do not need to be synchronized for the hospital scenario.

Patients might be allowed to specify transitive user policies, whereas doctors should not be able to forward an access right given to them in a user policy. Room policies should not be transitive.

University. In a university setting, students and faculty members specify their user policies. However, room policies are established by a central authority.

For an office, the authority is likely to give the right to establish its room policy to the occupant of the office. For lecture rooms and hallways, the authority typically would set the room policy such that room and user policies become synchronized. That is, upon receiving a room query, the location system consults the user policies of users in the room/hallway before returning their identity. For offices, user and room policies are typically not synchronized.

User policies might be transitive, whereas for room policies, the institution may decide not to let the occupant of an office transfer the right to other people.

4 Service Trust

As explained in Section 2, a location system can consist of multiple location services. Some of these services, such as the People Locator service shown in Fig. 1, do not generate their own location information. Instead, they process location information received from other services. To avoid information leaks, the location system must ensure that only services that implement access control checks are given location information.

One way to implement this condition is to require that the user or room policy grants a service access to location information before it is given this information. This option is appropriate for services that must be able to issue requests for location information. For example, the Device Locator service shown in Fig. 1 has to create queries for devices upon receiving a user query. However, for services such as the People Locator service, this option gives the services more privileges than they really need. The People Locator service only forwards requests received from clients. By granting it the right to issue requests, we increase its exposure in case of a break-in. If an intruder breaks into the service and begins issuing requests, the location system will grant access to these requests.

Due to these reasons, we introduce the concept of *service trust*. If a service, such as the People Locator service, is trusted, it is given location information, even if it is not allowed to issue requests. For a particular query, a service needs to be trusted by the policy maker that defines the corresponding user or room policy. Therefore, for a hospital, the set of trusted services should be defined by the central authority. For a university, each user and each owner of a room define their own set of trusted services. The trust assumption is that the service implements access control as follows:

In the first step, the service checks whether the policy maker has granted access to the entity issuing the request. Only if this check is successful, the service proceeds to the second step, else access is denied. In the second step, the service checks whether the entity from which it received the request corresponds to the entity that issued the request. If it does, access is granted. If it does not, the service has to verify whether the policy maker trusts the entity before access is granted. Else access is denied.

How can we verify whether a service fulfills its trust assumption? We require services to sign whatever location information they return to achieve non-repudiation. Therefore, an entity trusting a service can reactively identify misbehaving services and revoke the trust in them.

5 System Design

Based on our discussion in Sections 3 and 4, we now present the design of our access control mechanism for a people location system. We build on three main concepts. First, services respond to a location request only after performing a location policy check that verifies that the location seeker has access. Second, services verify that the service from which they receive a forwarded request is trusted before returning an answer to it. Third, services can delegate both location policy and service trust checks to other services; delegation can be used to eliminate redundant checks. In this section, we motivate and discuss these concepts.

5.1 Digital Certificates

For implementation purposes, we assume that location policy and trust decisions are stated in digital certificates. A digital certificate is a signed data structure in which the signer states a decision concerning some other entity. There are various kinds of digital certificates. Our implementation is based on SPKI/SDSI certificates [6], which we discuss in Section 6.1.

We introduce the following notation for expressing a policy or trust decision:

$$A \xrightarrow[scope]{type} B \ . \tag{1}$$

A is the entity making a decision concerning B. The type of decision is described above the arrow (*policy* or *trust*). In decisions of type *policy*, A gives B access to some location information, that is, these decisions express location policies. In decisions of type *trust*, A specifies that she trusts service B. A can limit the scope of a decision by stating the scope below the arrow (e.g., whose location policy is specified).

As shown before, there are two types of certificates: location policy certificates and trust certificates. When there is a request, a service must first check whether the location policy grants access to the issuer of the request. The service tries to build a chain of location policy certificates from itself to the issuer of the request. Each certificate in the chain needs to give the next entity further down the chain access to the requested location information. In addition, any constraints listed in the certificate (e.g., time or location based) must be fulfilled to have the policy check succeed. We elaborate on location policy certificates in Section 5.2. If a service receives a forwarded request, it must also check whether the service from which it got the request is trusted. Similar to the location policy check, the service tries to build a chain of trust certificates from itself to the forwarding service. We discuss trust certificates in Section 5.3.

5.2 Location Policy Check

If an entity is allowed to retrieve a user's location information, then there must be a certificate specifying whether and what type of information it can retrieve. (We limit the description of our design to user policies. Room policies are dealt with in a similar way.) For example, a certificate could state that Bob can locate Alice at a coarse-grained level. If Bob then tries to locate Alice, a service receiving his location request needs to check the existence and validity of a certificate permitting this access.

A location service does not have to be aware of the identity of the entities that have access rights. If Bob can prove that he is allowed (by presenting a digital certificate), he will be granted access. This solution makes dealing with unknown users easy. Solutions proposed earlier (e.g., by Leonhardt and Magee [5]) rely on policy matrices consisting of querying users and users whose location can be queried and thus rely on the system being aware of the identity of querying users.

In addition, some digital certificates (such as SPKI/SDSI certificates) can give rights both to single entities and to groups of entities. That is, with a single certificate, Alice can, for example, give all her friends access.

Depending on the environment, different entities issue certificates. For the two environments introduced in Section 3.6, certificates are specified as follows:

Hospital. We show some of the certificates issued in the hospital environment: (PL denotes the People Locator service, C the central authority, S the surgery ward, and D_S^C all doctors that are classified by C as belonging to S.)

$$(a) \quad PL \xrightarrow{policy} C \qquad\qquad (b) \quad C \xrightarrow[A]{policy} D_S^C \qquad\qquad (c) \quad D_S^C \longmapsto B \ . \quad (2)$$

First, the administrator of the People Locator service enables the central authority to decide about all the patients' location policies. Second, the authority gives all doctors in patient A's ward access to her location. Third, the authority certifies that doctor B belongs to the surgery ward. Note that this certificate is a membership certificate (denoted by the special arrow) and does not grant any rights on its own.

If B inquires about A's location, the People Locator service deduces that B has access to A's location since it can combine certificates 2(a), (b) and (c) in a chain of certificates and conclude

$$\mathrm{PL} \xrightarrow[A]{policy} \mathrm{B} \ . \tag{3}$$

In addition to certificate 2(b), the authority also issues a certificate that gives patient A the right to let additional people locate her. A can do so by issuing corresponding certificates.

University. In the university environment, the administrator of the People Locator service gives student or faculty member A access to her location information:

$$PL \xrightarrow[A]{policy} A \ . \tag{4}$$

A can define her own location policy by issuing additional certificates.

5.3 Service Trust Check

Trust decisions are also stated in digital certificates. As an example, we show how the Device Locator service handles trust decisions for the two environments introduced in Section 3.6. Typically, the Device Locator service is not directly contacted by users, but by other services (e.g., the People Locator service).

Hospital. The administrator of the Device Locator service (DL) lets the central authority specify the trusted services used for answering patient queries. The authority states that all services run by the hospital (T_C) are trusted. The authority also states that the People Locator service is in this set. The certificates look as follows:

$$(a) \quad DL \xrightarrow{trust} C \qquad\qquad (b) \quad C \xrightarrow{trust} T_C \qquad\qquad (c) \quad T_C \longmapsto PL \ . \tag{5}$$

Upon receiving a forwarded user query from the People Locator service, the Device Locator service combines these certificates and concludes that the People Locator service is trusted.

University. In the university scenario, users specify their trusted services. The Device Locator service thus gives student A the right to define this set. Assuming A trusts the People Locator service, the certificates look as follows: (T_A denotes the set of services trusted by A.)

$$(a) \quad DL \xrightarrow[A]{trust} A \qquad\qquad (b) \quad A \xrightarrow[A]{trust} T_A \qquad\qquad (c) \quad T_A \longmapsto PL \ . \qquad (6)$$

5.4 Delegation

Entities in our system grant other entities particular kinds of rights. For example, Alice grants Bob access to her location information. It is up to Alice to also grant Bob the right to forward the access right to a third entity. If she does so, Alice effectively delegates the right to decide about her location policy to Bob. In the remainder of this paper, we are going to use the term *delegation* whenever an entity grants access rights to a second entity and it also permits the second entity to grant these rights to a third entity.

In the rest of this section, we elaborate on delegating policy and trust decisions. In addition, we explain how users can delegate trust decisions to an organization.

Location Policy and Trust Checks. Each trusted location service has to implement location policy and trust checks. However, to reduce overhead or in the case of low processing power, not every service has to build the potentially long certificate chains itself. It can delegate this task to some other service. For example, the Device Locator service is likely to delegate location policy checking to the People Locator service since the People Locator service needs to build a certificate chain for each request anyway. After validating the chain, the People Locator service issues a new certificate that directly authorizes the location seeker. It gives this certificate to the Device Locator service, which thus does not have to validate the entire chain again.

Organizations. If there are lots of services available, it is cumbersome for a user to issue a certificate for each service that she trusts. We assume that trust in a service is closely tied to the entity that administers this service. For example, a user might trust all the services run by her company. Therefore, we give users the possibility to state in a certificate that they trust all the services in a particular organization. The organization then generates a certificate for each of the services that it runs. In this situation, a user effectively delegates the decision about which services she trusts to the organization, and she relies on the organization to do the right thing. For the university environment, certificate 6(b) could thus look as follows:

$$A \xrightarrow[A]{trust} T_C \ . \qquad (7)$$

5.5 Privacy

Location policy and trust certificates can be stored in a database from which anyone can retrieve them. However, since location policies contain personal information, policy makers might want to restrict access to the certificates. We now discuss two methods to limit their exposure. The first method keeps policies secret from location seekers, the second one from the location system. Both methods do not require any changes to the access control mechanism of the location system, they differ only in the way the mechanism is deployed. Note that a location seeker can obtain some policy information implicitly by issuing queries to the location system. If the location seeker is denied access, it concludes that the policy forbids access. Similarly, by analyzing the returned location information, it might be able to deduce that it has only coarse-grained access.

For both methods, we assume that each policy maker has a trusted host where it keeps its issued certificates. If the policy maker is a central authority, the trusted host can be the authority itself. If the policy maker is an individual, the trusted host is a designated host. This host has to be available all the time so that it can provide certificates upon a request. An individual has to make a trade-off between its privacy concerns and the convenience provided by its chosen solution. If the individual is willing to trust an organization, it can have the organization run a centralized repository for it. If it is not willing to trust an organization, it needs to set up its own repository.

To prevent policy information from leaking to location seekers, we have the trusted host control access to the certificates stored with it. Only location services entitled by the issuer of a certificate are allowed to access a certificate. This method increases the load on a location service since the service can no longer require location seekers to provide any certificates required for answering the request. Instead it has to retrieve them itself. Delegation or storing certificates in an encrypted form could reduce some of the load on a location service.

To prevent policy information from leaking to the location system, a policy maker has all queries go through its trusted host, which runs access control. The certificate chains expressing the location policy are rooted at the trusted host. If the access control check succeeds, the trusted host issues a request to the People Locator service. The trusted host is the only entity that has access to the location information offered by the People Locator service. A variant of this method is to implement the People Locator service in a completely distributed way. That is, each policy maker runs its own People Locator service on its trusted computer. This solution is similar to the one introduced by Spreitzer and Theimer [4]. However, running and administering a complete People Locator service is a potentially heavyweight operation.

5.6 Discussion

Let us summarize the main advantages of our location system:

No bottleneck. Certificates do not need to be kept at a centralized node, and there is no bottleneck through which each request has to go. The services perform access control independently of each other and approve a request only if it is supported by certificates.

Support of unknown users. The system does not need to know the identity of location seekers. All the system requires is a digital certificate that grants access to the entity.

Support of group access. With a single certificate, an entire group of entities can be given access to someone's location information.

Transitivity control. SPKI/SDSI certificates allow handing out non-transitive access rights. In a valid certificate chain, all but the last certificate in the chain need to allow transitivity of access rights.

We have been able to use digital certificates for expressing location policy and trust decisions in both of our example environments. Therefore, the mechanism for controlling access to location information is identical. Similarly, we can use the same tools for building and proving certificate chains in both environments. However, setting up these mechanisms and tools is environment-specific. We now discuss some of the differences.

User interfaces. Most probably, there are going to be different user interfaces in different environments. In the hospital case, there is a strong emphasis on being able to define groups and assigning them access. In the university case, users are more likely to authorize other people directly. Though the actual interfaces might differ, they would both create location policy (and trust) certificates in the background.

Types of chains. The types of chains checked upon a query also depend on the environment. Either a service checks only the user or room policy chain or it checks both chains. For example, when processing a room query about the people in a lecture room in a university, the location system first checks the room policy of the room. If this check succeeds, the system will check the user policy for each of the people in the room. The system returns only location information about people for whom this latter check succeeds.

6 Implementation

6.1 Digital Certificates

To implement the policy and trust decisions introduced in Section 5, we rely on SPKI/SDSI certificates [6]. Authentication and authorization based on these certificates does not require a global naming structure. Instead, the authentication and authorization step are merged, and a certificate gives access rights directly to a public key. There are tools that evaluate chains of SPKI/SDSI certificates and that decide whether requests should be granted access.

In the example below, we show a SPKI/SDSI certificate in which Alice (`issuer`) gives Bob (`subject`) access to her location information (i.e., $A \xrightarrow[A]{policy} B$ using our notation). The type of access is described after the keyword **tag**.

Note that the certificate has to be accompanied by Alice's signature (not shown here).

```
(cert (issuer (public_key:alice))
      (subject (public_key:bob))
      (propagate)
      (tag (policy alice
         (* set (* prefix world.cmu.wean) world.cmu.doherty.room1234)
         (* set  (monday (* range numeric ge #8000# le #1200#))
                 (tuesday (* range numeric ge #1300# le #1400#)))
         coarse-grained)))
```

Alice gives access to Bob's public key (denoted by `public_key:bob`). The keyword **propagate** states that Bob is allowed to give the granted right to some other entity by issuing another certificate. If Alice decided not to let Bob forward his access right, she would omit this keyword. Bob can locate Alice only if she is either in Wean Hall or in Room 1234 in Doherty Hall and on Monday between 8am and 12pm and on Tuesday between 1pm and 2pm. Also, Bob can locate Alice only at coarse-grained granularity.

6.2 Location System

We have implemented a subset of the location system shown in Fig. 1. The system consists of the People and Device Locator services and location services that proxy to several calendar systems, that locate devices connecting to CMU's wireless network, and that exploit login and activity information.

We use SSH at the transport layer to achieve mutual authentication of entities and confidentiality and integrity of information. Access control is entirely based on SPKI/SDSI.

Location information and SPKI/SDSI certificates are transmitted between services using the Aura Contextual Service Interface [7], which is a protocol running over HTTP and which exchanges messages encoded in XML. Our SPKI/SDSI implementation has been implemented in Java and is based on previous work [8]. There is an HTML-based front end to the service that lets users specify location policies and conduct searches. The front end makes dealing with certificates transparent to users. Currently, the certificates are stored in a centralized database.

In our system, if Alice directly gives Bob access to her location information in a certificate, it takes about 0.6 sec on a Pentium III/500 to add this certificate to the certificate chain for Alice and to verify the entire chain upon a request.

7 Related Work

Several location systems [2, 3, 9, 10] have been proposed. They are based on only one location technology, are implemented only within one administrative entity, and/or do not address the various access control issues mentioned in this paper. We discuss two notable exceptions:

Spreitzer and Theimer's location system [4] is based on multiple technologies. Each user has her personal agent that gathers location information about her and that implements access control. The authors design the system to work in an environment with different administrative entities, although the actual implementation runs only within a single entity and the authors do not mention how users specify trusted services. A key difference from our system is that users in a room, not the owner of a room, determine the room policy of this room.

Leonhardt and Magee [5] argue that user and room policies need to be consistent. They propose an extension to the matrix-based access control scheme. We believe that having consistent user and location policies is difficult to achieve when policies are established independently of each other. The authors do not discuss how policies are established in their system.

There has been some previous work about authorizing intermediate services, for example, Howell and Kotz's quoting gateways [8]. Howell and Kotz focus on intermediate services that create new requests upon receiving a request and thus need to be authorized to issue requests. However, in scenarios like ours where some services only forward requests, this model would give too many capabilities to intermediate services. Our model of trust avoids this risk.

The Instant Messaging / Presence Protocol working group [11] proposes to return wrong information instead of denying access to information. We refrain from such a solution since it erodes trustworthiness into the location system. Also, if a location seeker tried to validate the wrong location information, it could still conclude that it was actually denied access. As an alternative to the proposed method, we suggest that policy makers build wrapper services that access the location system and that translate "access denied" messages. For example, the wrapper service could return a busy signal when a user temporarily does not want to be disturbed and thus decides not to be locatable.

The Location Interoperability Forum [12] and the Geopriv working group [13] discuss privacy aspects of user queries. The former suggests that location seekers should not be able to learn about the content of location policies. The latter proposes to protect the identities of both location seekers and located entities. We have presented a mechanism for hiding location policies and the identity of location seekers in Section 3.5. Protecting the identity of located entities strongly depends on the actual location service. For example, a user can buy a prepaid phone card when using a GPS-enhanced phone and hide his identity from the GPS location service.

8 Conclusions

We have analyzed the challenges in controlling access to people location information and presented the design of an access control mechanism. The design relies on several key concepts: policy certificates for expressing location policies, trust certificates for dealing with services belonging to different organizations, and delegation for avoiding redundant access control checks and to lower the burden on users. We have shown feasibility of our design with an example implementation, which we are currently deploying at Carnegie Mellon.

We have been able to formulate all of our policy and trust decisions using SPKI/SDSI certificates. These certificates provide a high degree of flexibility. A ubiquitous computing environment poses new challenges on access control that cannot be easily satisfied by conventional mechanisms. We believe that due to their flexibility, SPKI/SDSI certificates are a promising approach and deserve further investigation on their usability in such environments. Namely, these certificates could potentially not only protect access to people location information, but also to other kinds of information that is gathered in a similar way as people location information.

Acknowledgments

We thank Glenn Judd and the anonymous reviewers for their comments. This research was funded in part by DARPA under contract number N66001-99-2-8918 and by NSF under award number CCR-0205266. Additional support was also provided by Intel.

References

1. Garlan, D., Siewiorek, D., Smailagic, A., Steenkiste, P.: Project Aura: Towards Distraction-Free Pervasive Computing. IEEE Pervasive Computing **1** (2002) 22–31
2. Bahl, P., Padmanabhan, V.: RADAR: An In-Building RF-Based User Location and Tracking System. In: Proceedings of IEEE Infocom 2000. (2000) 775–784
3. Priyantha, N., Chakraborty, A., Balakrishnan, H.: The Cricket Location-Support System. In: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000). (2000)
4. Spreitzer, M., Theimer, M.: Providing Location Information in a Ubiquitous Computing Environment. In: Proceedings of SIGOPS '93. (1993) 270–283
5. Leonhardt, U., Magee, J.: Security Considerations for a Distributed Location Service. Journal of Network and Systems Management **6** (1998) 51–70
6. Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., Ylonen, T.: SPKI Certificate Theory. RFC 2693 (1999)
7. Judd, G., Steenkiste, P.: Providing Contextual Information to Ubiquitous Computing Applications. To appear in *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)* (2003)
8. Howell, J., Kotz, D.: End-to-end authorization. In: Proceedings of the 4th Symposium on Operating System Design & Implementation (OSDI 2000). (2000) 151–164
9. Harter, A., Hopper, A.: A Distributed Location System for the Active Office. IEEE Network **8** (1994) 62–70
10. Ward, A., Jones, A., Hopper, A.: A New Location Technique for the Active Office. IEEE Personal Communications **4** (1997) 42–47
11. Day, M., Aggarwal, S., Mohr, G., Vincent, J.: Instant Messaging / Presence Protocol Requirements. RFC 2779 (2000)
12. Greening, D.: Location Privacy. http://www.openmobilealliance.org/lif/ (2002)
13. Cuellar, J., Morris, J.B., Mulligan, D.: Geopriv requirements. Internet Draft (2002)