Security Considerations for a Distributed Location Service

Authors:

Ulf Leonhardt and Jeff Magee

Affiliation:

Department of Computing

Imperial College of Science, Technology and Medicine

London, UK

Contact Details:

Department of Computing

Imperial College

180 Queen's Gate

London SW7 2BZ

United Kingdom

Email: U.Leonhardt@ic.ac.uk

Tel: +44-171-594 8249

Fax: +44-171-581 8024

Suggested running head:

SECURITY CONSIDERATIONS FOR A DISTRIBUTED LOCATION SERVICE

Date:

Friday, September 19, 1997

Abstract

Mobile computing, wireless communications, and cheap location tracking and navigation systems have made location data a valuable and available commodity for many different kinds of computing applications. However, there are fears that this new wealth of personal location information will lead to new security risks, to the invasion of the privacy of people and organisations. In this paper, we discuss security requirements faced by a location service in different organisational contexts. We argue that fine-grained access control requires a symbolic location model over which access control is specified. We outline the salient features of a location service supporting such a location model. The two main classical security models, Lampson's access matrix and Bell-LaPadula's security labels, are analysed with view to their application to location information. We argue that those schemes need to be generalised to deal with multiple targets in order to be applicable to location information. Based on the generalised models, we propose a concrete security model for location information which protects both personal and organisational privacy. We have implemented this model over a prototype implementation of a general location service.

Keywords: *location service, access control policies, discretionary access control, mandatory access control*

1. Introduction

Security of information systems are of great concern to individuals and organisations, though for different reasons. Commercial organisations are mostly interested in the integrity of their data. The military worries more about secrecy (see [2] for a discussion). Individuals are concerned about privacy, which is broadly defined as personal control over the secrecy of private information.

In this paper, we consider the security requirements of a particular class of information services: a location service [8]. A *location service* provides information about the physical position of located-objects. The category of *located-objects* comprises all physically mobile objects which can either be tracked or which can determine their own position. This includes people, computers, telephones, cars, and others. A location service provides two basic functions: enumerating all the located-objects at a given location, and enumerating all the known locations for a given located-object.

The spectrum of applications for a location service is wide. It ranges from mobile telecommunication systems to emergency assistance services and computer-supported cooperative work. As a result, location services will often become repositories of potentially sensitive personal and corporate information. *Where you are* and *who you are with* are closely correlated with *what you are doing*. To leave this information unprotected for everybody to see is

clearly undesirable. People would feel uncomfortable if their every move could be watched anonymously. Similarly, businesses would probably not like the idea of competitors or staff monitoring the attendance of every meeting. Further, location data will be used, directly or indirectly, as input for decision-making processes. Hence, also the integrity of location data is important.

We conclude that location information needs to be protected against unauthorised disclosure and modification. However, the exact level of protection varies widely from context to context. Personal location services, corporate location services, and military location service will all have different requirements for secrecy and integrity. Hence, we concentrate in this paper on models for specifying security. Those models can then be used to address the requirements of a specific location service.

In this paper, we shall focus on the secrecy aspects of security. We outline two typical deployment scenarios for a location service. Then, we explore the application of traditional mandatory and discretionary security mechanism to our problem. We conclude with a brief description of our prototype implementation, along with a discussion of related work.

2. Requirements

In this section, we outline two usage scenarios for a location service. These scenarios highlight different deployment environments and the resulting sets of requirements. We are especially concerned with the balance between security imposed by the system (mandatory security), and security specified by individuals (discretionary security).

2.1. Scenario I: Intra-organisational Location Service

Within organisations, there is often the need to locate people in real-time. For example, trucking companies often use GPS-based location systems to efficiently reroute vehicles when goods need to be picked up on short notice [3]. Further examples of the use of a location service within a organisation include computer-supported collaborative work, communication with mobile workers, and location-based security mechanisms. The common theme in this scenario is that acquisition, management, and use of location information are ultimately controlled by a single decision-making body. We refer to this case as the *intra-organisational scenario*.

The security policy is set by the organisation for the whole location service. Typically, local discretion is permitted only within the bounds defined by the organisational policy. The system is closed to outside access (except for limited and controlled cases). The people and mobile objects that are to be tracked are registered with the organisation. The coverage area, however, may well be very large if the tracking technology and the communication network

permit.

In this setting, both integrity and accuracy of location information are of importance, since the organisation's processes and decisions will be affected. At least, it should be possible to tell whether information is trustworthy. Some applications, as location-based authorisation systems, demand a very high degree of trustworthiness from the location information. Other applications may well trade availability of information against accuracy.

As far as secrecy within organisations is concerned, we see two basic requirements: location-centric privacy and user-centric privacy. Firstly, the organisation may want to allow only a certain group of people to find out who is in a specific room or building. For example, a floor of a building might be 'open' to all the people who work there, but not to people from other floors and buildings. Secondly, a person's location should probably only be visible to a restricted group of other people. For example, the managing director might be visible all the time to his or her secretary. Other people can see him only when he is in his office.

Note: User-centric privacy is not the same as personal privacy, but rather user-centric organisational privacy. Personal privacy is an orthogonal concept.

At present, most location services fall into the intra-organisational category. However, provision of a global and public location service will require a more general, inter-organisational approach.

2.2. Scenario II: Global Location Service

In contrast to the well-controlled, relatively closed environment described above, we now discuss the scenario of a global location service. We expect that such as service would be provided by a network of loosely cooperating providers, very similar to today's mobile telephone system. Customers would subscribe to one or more service providers. The providers would have roaming agreements with each other. Subscription would be necessary in order to be tracked by the service, and also to access the service. Service level and security provisions would be governed by a contract of law.

The applications for such a service are the same as described in intra-organisational scenario. However, a global location service makes those applications a much more practicable proposition to small organisations and private users. Further, there is scope for third-party location-aware services. For example, such a service might be responsible to automatically inform emergency services when a distress signal from a subscriber is received. On the other hand, users will often have to trust the service providers to obey the security policy laid down in the service contract.

The global service scenario represents an open system in two ways. Firstly, roaming subscribers may encounter

service providers which they have not met before. Secondly, service providers may encounter unknown subscribers in their area. There could also be competing service providers in the same area, thus further complicating the situation. The problem here appears to be mainly one of cross-domain authentication and user profile management, which is outside the scope of this paper.

We envisage that service providers would be obliged to implement certain generic security policies, such as non-disclosure to unauthorised third parties. Additionally, each subscriber would specify an acceptable security policy for him or her. For example, somebody might choose to be visible to his boss at work but not at home and not on weekends. These policies would presumably be mostly user-centric, while location-centric policies (for example, to protect the privacy of a person's home) could be useful, too.

Ideally, there should also be more generic ways to specify access authorisation. For example, when attending a conference I would like to be visible to all the other attendees without actually knowing them. Similarly, I might wish to be anonymous in locations matching a given constraint, such as a motor-way.

Generic authorisation constraints are especially important since the service is partitioned among many providers. These providers must rely on local knowledge to make access control decisions. Constraints that require frequent access to non-local information cannot be considered a scalable solution to this problem.

The requirements governing integrity and accuracy of the location information can also expected to vary widely. Even a single subscriber could have multiple accuracy requirements for different applications.

2.3. Summary

In both scenarios, secrecy is the main concern. In scenario I, secrets of the organisation need to be protected while individual's privacy is of lesser concern. In scenario II, subscribers' personal privacy is the main requirement. In both cases, privacy has user-centric and location-centric components.

A location service also needs to be protected from false location data. Further, there is a need in both scenarios to distinguish trusted from untrusted location information. The integrity of trusted information needs to be protected against improper modifications.

In the remainder of this paper, we shall focus on models for the specification of secrecy constraints that are applicable to both scenarios.

3. Location Domains

Following the approach of management domains [13], we have in [8] proposed location domains as location-de-

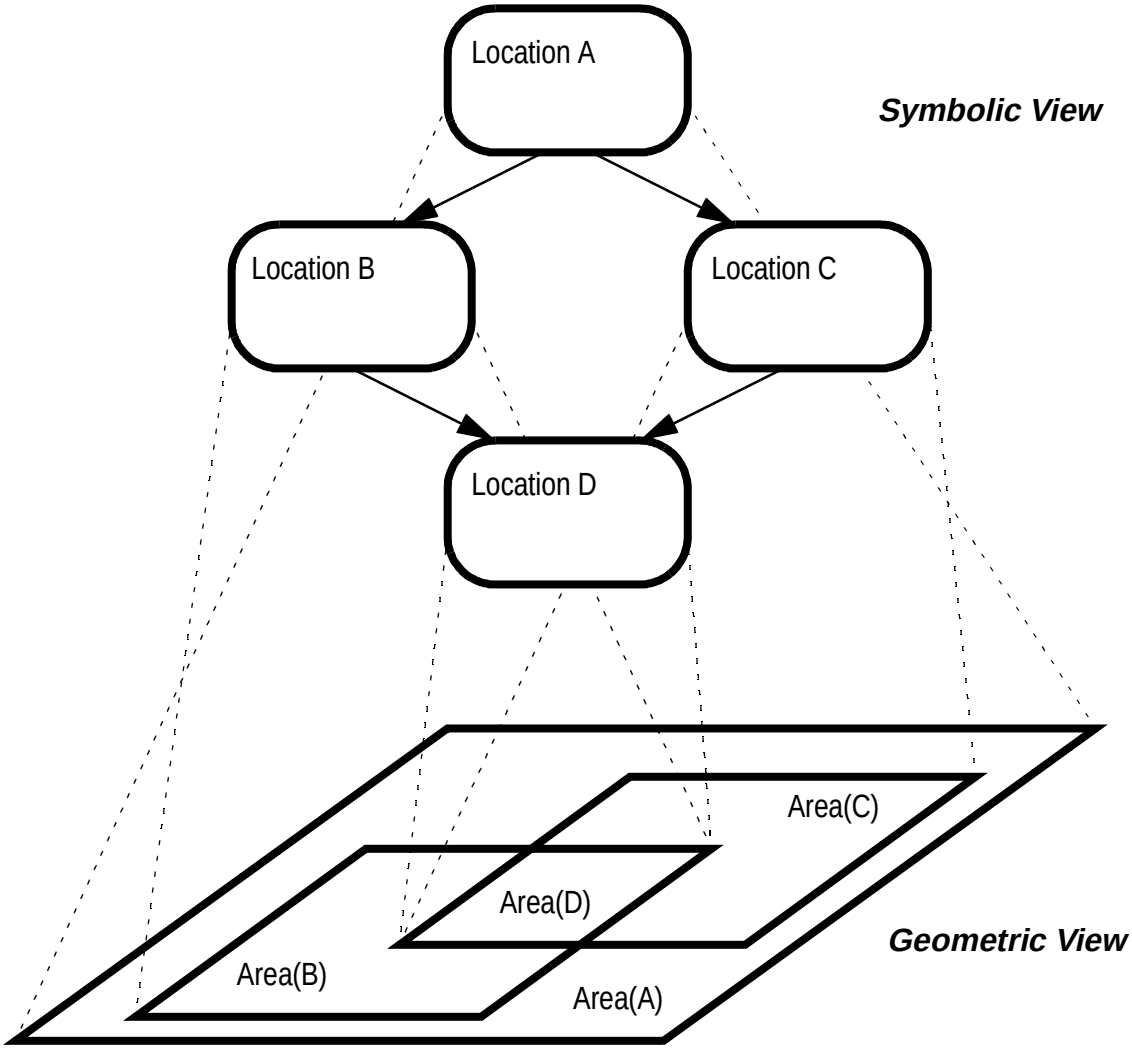

Figure 1: Hybrid location model

pendent grouping mechanism for located-objects. Location domains are explicitly defined and bound to a well-defined geographical area. Location domains are either defined as a sub-domain of another location domains, or associated with a located-object. We use the following naming convention for location domains:

```
[<abstract area>|<geographic area>] @ [<area>:<label>|<located object>]
```

Table I shows some examples of this naming convention.

`Rm@/Hux:449`	This domain corresponds to Room 449 in Huxley Building
`Level@/Hux:4`	This is Level 4 in Huxley Building
`>4m@/Students/Joe`	This is a circle of radius 4 meter around Student Joe
`Rm@/Students/Joe`	This is Joe's current room

Table I: Examples of location names

Location domains provide a named representation of certain geographical areas. If a located-object happens to be at the same time in the same area, it automatically and implicitly becomes a member of the location domain.

The set of location domains in the system is referred to as the location domain space, which is separate from the management domain space. The location domain space is partially ordered by domain composition. The ordering of the domain space is isomorphic to the spatial containment relation between the areas associated with the location domains. This duality is illustrated by Figure 1.

Since a domain's geographical area may change over time, the structure of the location domain space changes. Typically, there are well-defined sub-sets which remain fixed for physical reasons. For example, rooms in one building do not normally move to another building. Presently, we are investigating whether effective management policies over the location domain space can be specified using only those location domains with a fixed position.

4. Location Service

We define a *location service* as a shared object that provides information about the physical location of located objects. It encapsulates resources for location tracking and positioning. Sometimes, also historical location data and geographic information are provided.

Special-purpose location services are part of mobile telephone systems (such as GSM [10]) and computer-aided fleet dispatch systems (see [3]). Researchers have designed and implemented small-scale general location services (examples include[14][11][8]). A general location service facilitates context-aware applications [12], location-dependent service hand-overs [6], and location-aware services in general. We expect that location services will be-

come increasingly important as mobile computing devices grow in sophistication and popularity.

Functionality and manageability of a location service depend primarily on the location model used. We argue that manageability (and security) require symbolic location abstractions. One such abstraction is the location domain model described in section 3. In practice, we expect the symbolic model to be complemented by a geometric model such as a three-dimensional coordinate space. Here, we shall concentrate on the symbolic model.

A location service over a symbolic location model offers the following basic functionality:

- *given a located-object, return all the current symbolic locations of this object*
- *given a symbolic location, return all the located-objects currently located there*

The difficulty in specifying a security model over these two functions is indicated by their symmetry. Either function can reveal all available location information. Hence access control for both functions must be consistent.

As far as the architecture of the location service is concerned, many different processing and distribution models are possible. However, in this paper we shall focus on architecture independent security models.

In the following sections, we are going to discuss how security for a symbolic location service with those two functions can be specified. We expect that the results can be applied to location models including geometric data and to location services with more complex functionality.

5. Why is access control to location information different?

Location information essentially consists of fast-moving dynamic relationships between multiple objects. Difficulties for existing approaches are the dynamism of the information, and the fact that location information does not consist of knowledge about objects, but of knowledge about relationships between objects.

The first difficulty mentioned above arises because management systems tend to rely on a relatively static structuring of the problem domain, an example being the domain-based management framework described in [17]. There, the problem domain is structured into a graph of management domains, with each domain containing a set of references to managed objects. Managers are expected to explicitly add objects or remove objects from a domain. We believe that while the domain graph should remain mostly static, dynamic location-dependent domain-membership is required to manage mobile objects. However, this is more an architectural problem and lies outside the scope of this article.

The second difficulty mentioned above is the actual motivation for writing this paper. We seemed to be unable to specify security policies for location information using the standard models for access control, Lampson's access

matrix [7] and Bell-LaPadula's security labels (see [1]).

Traditionally, the use of access control is either mandatory (imposed by the system), or discretionary (left to the owners of the objects). Both approaches are based on the subject-target paradigm. In mandatory access control, a subject is allowed read access or write access to an target if certain axioms over the security labels of subject and target are satisfied. When using discretionary access controls, an access matrix (Lampson [7]) with possible subject-action-target combinations is constructed. Access by a subject to a target with an action is granted if the corresponding combination is a member of the access matrix.

With location information, there is no obvious target object. If the located-object is treated as the target object, it becomes very hard to specify a access control for all objects at a given location. If the location is made the target object, it is difficult to specify access control for a given located-object. Using both methods in combination is not satisfactory because the access control information would be duplicated.

In the remainder of this section, we describe how the classic access control models can be generalised to cope with this problem. For the purpose of this discussion, we shall assume a domain-based framework with dynamic location-dependent domain membership.

5.1. Matrix-based access control

Review In domain-based frameworks, matrix access control policies are specified by rules of the form:

```
<subject scope>   {list of actions}  <target scope>
```

Semantically, such a policy allows any subject from `<subject scope>` to perform one of `<list of actions>` on a target from `<target scope>`. This corresponds to an access matrix where both subjects and targets are domains. This additional level of indirection allow for policies to be specified over groups of objects rather than individual objects.

Application As far as location information is concerned, a typical (informal) policy is

```
Joe may see that Fred is located at Building@/School
```

A policy with the same meaning is:

```
Joe may see that Building@/School encloses Fred
```

Clearly, both policies specify the same thing - authorisation that Joe is allowed to observe a particular relationship,

collocation, between `Fred` and `Buidling@/School`. However, such a policy cannot be expressed adequately in the conventional subject-action-target paradigm. This limitation can be somewhat alleviated by using policies with additional constraints [9]. By using constraints, we can actually express the required policy in a canonical form:

```
Joe  {testForColocation(PERSON)}  Building@/School  WHEN PERSON=Fred
```

This specifies that Joe is allowed to perform the action `testForColocation(PERSON)` on `Building@/School` when `PERSON` equals `Fred`. The action, here `testForColocation(PERSON)`, contains Fred as an implicit target. Unfortunately, this necessitates the evaluation of the `WHEN` clause at run-time. The WHEN clause contains an arbitrarily complex first-order logic expression, which makes a light-weight implementation somewhat difficult. Even worse, conceptual clarity is lost. The essence of the actual policy is obscured: granting authorisation for an action that, symmetrically, affects the rights of multiple targets.

To deal with this problem, we propose the use of multi-target policies of the form:

```
<subject>  {action}  <target 1>...<target n>
```

The (informal) semantics of such a policy is: subject is authorised to perform action over the composite entity consisting of target 1 to target n. Obviously, multi-target policies are only useful for actions that affect multiple targets at the same time, such as binding of component interfaces in a distributed system by a third party, or brokering of deals. Applied to our example, this reads:

```
Joe  {testForCollocation}  Fred,  Building@/School
```

For completeness' sake, multiple subjects can also be introduced:

```
<subject 1>...<subject m>  {action}  <target 1>...<target n>
```

Multi-subject authorisation policies would describe authorisations for actions which require multiple subjects to perform an action together, such as opening a deposit locker, or authorising a cheque. A set of policies over m subjects and n targets corresponds to an $m + n$ dimensional access matrix. An example:

```
Sweden, Finland  {mediate}  Israel,  Syria
```

This policy specifies that `Sweden` and `Finland` may (together) `mediate` between `Israel` and `Syria`. This

does not convey authorisation for either `Sweden` or Finland to mediate alone.

Such policies are necessary for actions that operate over dynamic relationships between a fixed number of objects. The objects in those relationships can collectively act as subject or target of an action. Arguably, the relationships themselves could be promoted to first class objects, leading to an even more general solution. However, we believe in our context the additional complexity would not be justified.

Note: The approach of multiple source scopes and target scopes is distinct from the additional grantee scope proposed in [18]. This scope is used to specify object which a policy can be delegated to. Grantee-scopes are an extension facilitating the management of policies rather than extending their expressive powers. The approach described here is orthogonal and could be combined with grantee scopes.

5.2. Label-based access control

Mandatory access control in the domain framework is implemented by assigning security labels to management domains. All objects within a domain inherit the domain's label. If an object is a member of multiple domains, it inherits the least upper bound of all its parents' labels. Access is granted whenever the security labels for subject and target satisfy a certain set of axioms.

Analogously to the matrix-based case, the pair of a single subject and a single target objects alone does not contain enough information to decide whether access to location information should be allowed.

Hence, the security labels for both targets, that is location and located-object, should be consulted along with the label of the subject. Therefore, the axioms must cater for multiple subjects and multiple targets.

Review A label consists of a fixed number of attributes. There is an equivalence relationship defined over the set of values for each attribute. Further, attribute values may be partially or totally ordered. These relationships are used by the axioms to establish a 'dominates' relationship between labels. This in turn is also a partial ordering relation.

The most common label format, as used by Bell-LaPadula (see [1]), has two attributes. The first attributed S, a sensitivity level, is totally ordered. The second attribute C, a set of categories, is partially ordered by sub-set inclusion. A label A is said to dominate label B if both of A's attributes are greater or equal B's corresponding attributes

$$dominates(A, B) \Leftrightarrow (S_A \geq S_B) \wedge (C_A \supseteq C_B)$$

For the sake of brevity, we shall use the object's name to refer to the object's label in the 'dominates' predicate. The commonly used axioms are:

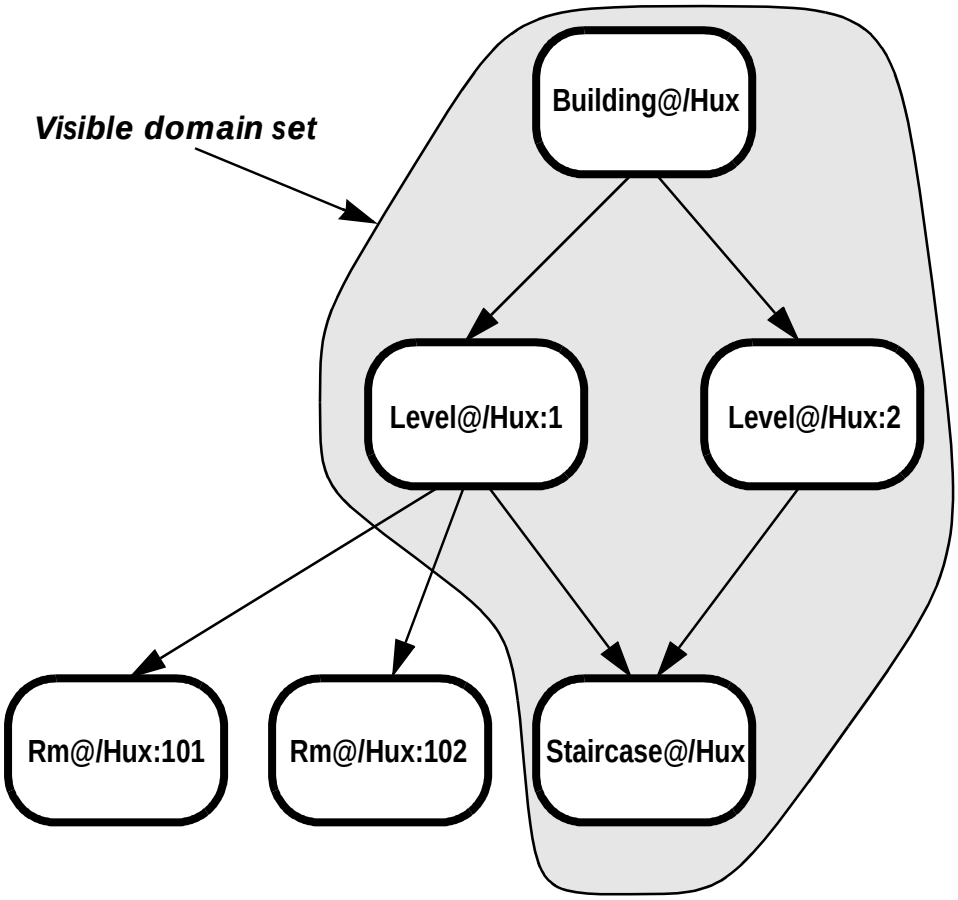

Figure 2: Location hierarchy with a visible domain set

- *Subject S may read target T only if* $dominates(S, T)$
- *Subject S may append target T only if* $dominates(T, S)$
- *Subject S may overwrite target T only if* $dominates(T, S)$ *and* $dominates(S, T)$

These axioms ensure that information may only flow from objects with lower security classification to objects with higher classification. Thus classified information cannot be declassified by 'normal' operations.

Application We need to define a set of axioms of over subject and target that allow a decision to be made whether access should be granted. In contrast to the approach described above, here we need to deal with two targets: locations and located-objects.

We wish to express the following high-level policy for mandatory access control:

`Location data maybe disclosed only if the secrecy of neither the located-object nor the location is infringed.`

In this context, 'infringement of secrecy' translates to a flow of classified information to a target with lesser classification.

We attach security labels S, L, and O to subject, location, and located-object, respectively. The above policy can then be expressed as follows:

`S may see L at O only if` $dominates(S, L)$ `and` $dominates(S, O)$`.`

The 'dominates' relationship is a partial order because the attributes values are drawn from partially ordered sets. Therefore, instead of verifying the *dominates* relationship for each label in turn, we may choose to compound all the target labels into a single label. The compound target label's level is intended to be greater or equal then any of the individual labels. This notion corresponds to the mathematical concept of a least upper bound (l.u.b) $lub(S)$over a set *S* partially ordered by the *dominates*-relationship. We define the l.u.b. over a set of labels as follows:

$$ub(S, x) \Leftrightarrow (\forall y \in S)(dominates(x, y))$$
$$(lub(S) = x) \Leftrightarrow ub(S, x) \wedge \neg(\exists z)(ub(S, z) \wedge dominates(x, z) \wedge \neg(x = z))$$

In this definition $ub(S, x)$ is a predicate that is true if *x* is an upper bound of set *S*.

Using the l.u.b. of the participants' security labels, we can express the above policy as:

`S may see L at O only if` $dominates(S, lub(\{L, O\}))$`.`

The computation of the l.u.b. can be simplified because the ordering of the security labels is based on the ordering of the labels' attributes: sensitivity level and category set. The l.u.b of two composite labels can be constructed

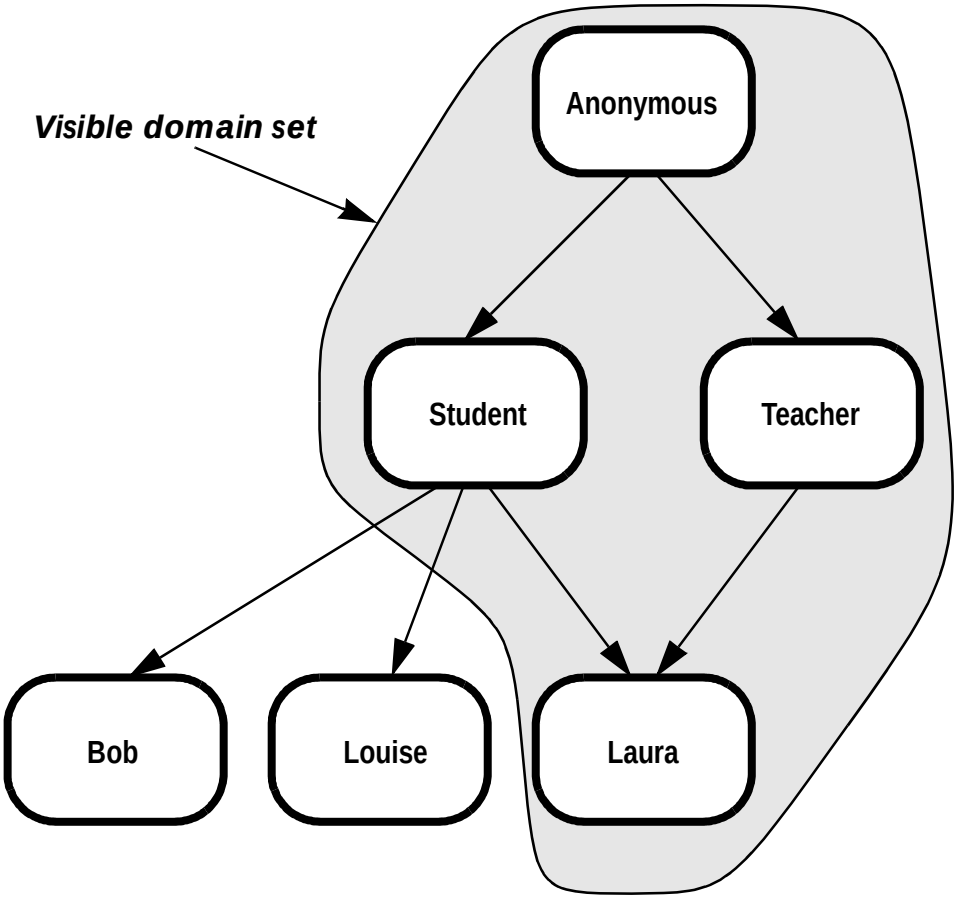

Figure 3: Located-object hierarchy with a visible domain set

from the least upper bounds of the corresponding attributes of the label. That is, we compute the l.u.b. of the sensitivity levels and the l.u.b. of the category set. The label consisting of both results is the l.u.b. of the two original labels. More generally, the least upper bound of a set of attribute tuples can be computed as the tuple of the least upper bounds of the individual attribute values:

$$lub(\{(a_1, \ldots, a_n)_A, (b_1, \ldots, b_n)_B\}) = (lub_1(\{a_1, b_1\}), \ldots, lub_n(\{a_n, b_n\}))_{A+B}$$

Note that the l.u.b. for each set of attribute values operates over the partially ordered set specific to that attribute. Axioms over operations with multiple subjects can be defined analogously. Here, the compound label of all subjects should be less or equal to the individual labels. Hence, the compound label is defined as the greatest lower bound (g.l.b) of the subject labels.

Consider the following policy with *T* and *S* defined as sets of objects:

$$S \text{ may } \texttt{read}\ T \text{ if } dominates(glb(S), lub(T)) .$$

This policy permits a flow of information from a group *S* of object to a group *T* if object provided that the least classified element of *S* still *dominates* the highest classified member of *T*. This shows how the Bell-LaPadula security model can be applied to actions with multiple subjects or multiple targets. Hence, Bell-LaPadula in a generalised form can be applied to the problem of specifying security for location information.

6. Mandatory versus Discretionary Access control

From a functional point of view, mandatory label-based access control provides a simple framework for consistent protection of secrecy. However, individuals cannot selectively allow or restrict access to private information. Hence, label-based access control is not a suitable mechanism to protect people's privacy. It will therefore be favoured by organisations with strong security requirements from the scenario I background. This simplicity, however, comes at the cost of reduced flexibility. Most environments from scenarios I and II will require some kind of matrix-based access control. These controls can be specified at the organisational level (scenario I), or by the owner of the information (scenario II).

From an administrative point of view, mandatory labels require a central authority for creating and assigning security labels. Therefore, a mandatory scheme is not suitable for decentralised environments, such as described in scenario II. Matrix-based systems can be administered either centrally (scenario I) or in a decentralised way (scenario II). In general, central administration is less complex but not always practicable.

We believe that few environments will rely solely on mandatory label-based access control to location information.

The needs of most scenarios can be satisfied with matrix-based access controls, perhaps using management domains and a policy notation as supporting framework. Further, security labels can be emulated by policies. The converse does not hold.

7. What kinds of access control policies are needed?

No single monolithic access control scheme appears to work well in all or even most of the typical deployment environments. Therefore, we advocate a mix-and-match approach which allows for orthogonal fine-grained access control mechanisms to be chosen and combined according to the actual security requirements. Additionally, conventional access control mechanism can be applied to protect the location service as a whole.

Our access control mechanism is structured into three layers: control of access, control of visibility and control of anonymity. In an actual system, only one or two of these layers might be used. Semantically, the authorisation granted by one layer is only a necessary precondition for the actual access authorisation. It can be overridden by higher layers. In the following paragraphs, we describe the functionality of each of the three layers.

Access policies specify the traditional level of access authorisation. That is, unauthorised queries are rejected. However, in order to achieve fine-grained access control also the query results need to be considered. A single query can produce a perfectly authorised result in one set of circumstances, and an unauthorised result in a different set of circumstances. Therefore, results which are unauthorised need to be removed from the result set. Only queries which cannot possibly produce authorised results should be rejected straight away. The decision whether a query should be rejected must not allow any inference regarding the affected locations and located-objects. Therefore, this decision should be made without reference to the current locations of located-objects.

```
Joe  {accessCollocation}  Fred,  Building@/Hux
```

This policy states that `Joe` is allowed to observe collocations between `Fred` and `Building@/Hux` (including all sub-locations).

Access policies specify necessary pre-conditions which may be strengthened by policies governing anonymity and visibility. We think of access policies as the “iron fence” surrounding the “playground” of the visibility policies and anonymity policies described below.

Visibility policies control the level of detail released about the *location of particular located-object*. These policies will typically act as a filter and replace detailed location information with less detailed information. Such a substitution is made possible by the hierarchic structure of the location domain space as shown in Figure 2. A set of vis-

ibility policies for a given subject and located object defines a set of visible location domains.

```
Joe  {accessLocation}  Fred,    Level@/Hux:1
```

This policy states that a co-location between `Fred` and `Level@/Hux:1` (including its sub-locations) may be observed as `Level@/Hux:1`. The policy does not specify whether access is allowed or whether the identity `Fred` should be revealed.

Anonymity policies control the level of detail released about the *identity of a located-object* at a particular location. This is conceptually very similar to the visibility policies described above. Instead of using a hierarchy of locations, we employ a hierarchy of identities as shown by Figure 3. The ordering of identities for a given located-object reflects increasing anonymity of identification. Therefore, we can automatically replace concrete identities with more anonymous identities in a result set without affecting its correctness. A set of anonymity policies for a given subject and location defines a set of visible identities.

```
Joe     {accessIdentity}   Anonymous,    Building@/Huxley
Joe     {accessIdentity}   Fred,         Building@/Huxley
```

The first policy states that within Huxley Building (and its sub-locations), everybody should be visible as anonymous. The second policy specifies an additional but non-conflicting authorisation to `Joe` allowing her to see `Fred` as `Fred` in Huxley Building and all its sub-locations.

Higher-level policies. The three levels of access control can be combined in different ways to implement higher-level organisational or personal security policies. Examples include:

- *"Correlate publicity of location with anonymity"*. Such a policy corresponds quite closely to our intuition about privacy. In a public place, we expect to be anonymous, whereas everybody knows our identity when we are in our office. Such a style of access control can be specified directly using anonymity policies.
- *"Correlate enquirer's role with the revealed granularity of location"*. Actually, we would like to like to correlate the purpose of a query with the granularity of the result, but this is hard to do directly. Fortunately, the enquirer or his *role* are often a good approximation for the purpose of the query. This high-level policy can be refined using the visibility policies described above.
- *"Do not allow outside access"*. In type I scenarios, we expect this to be a common high-level policy. While this could be expressed using visibility policies or anonymity policies (or rather, their absence), we prefer not to over-complicate things. Access policies offer a simpler and thus more suitable mechanism to specify "hard"

access control (as opposed to “soft” access control with anonymity and visibility policies).

These examples show that the proposed policy types offer significant flexibility, thus enabling them to address the requirements of a range of organisational contexts from both scenarios I and II. Larger case-studies will be required to evaluate the practical suitability of our approach.

8. Prototype implementation

We have implemented a location service with access control mechanisms as described in section 7. As prototyping platform we have used a commercial object-relational database management system with a plug-in for three-dimensional spatial data types and operations (Informix’ Illustra [4] with 3D Spatial DataBlade [5]). Our prototype allows for real-time data feed of symbolic location sightings (provided by Active Badges, and the UNIX ruser service), and geometric sightings from GPS receivers. The database supports queries to located-objects and to locations via SQL and a Web-based front-end.

The hierarchic symbolic data model was implemented on top of the spatial data types provided by the Spatial DataBlade. The location domain graph is defined implicitly by the spatial contain-relationship available in the DataBlade module. All the logic is implemented using stored procedures in Illustra-SQL.

For access control we use a filtering approach, that is, each layer of access control is a stored procedure filtering unauthorised results from the query results. Filters are idempotent and can be chained. We have implemented filters implementing the access control policies (access, visibility, anonymity) as described above. The policies are also stored in database tables.

This prototype has allowed us to verify the consistency of our location data model and the security model. We have also learned that some parts of the location service, such as the location hierarchy and the mapping from coordinates to symbolic locations, need to be implemented outside the database for efficiency reasons. This also applies to the access control checks since they operate over the location hierarchy.

9. Related Work

Typical commercial location service implementations, as used by GSM [10] for example, need to offer strong guarantees for the secrecy of location data. Secrecy is ensured by closing the system to outside access and by coarse-grained traditional access control (if there is any). More sophisticated approaches have been proposed by the research community.

Researchers at Xerox PARC where among the first to recognise the security implications of a location service [15].

In [16] they argue that different environments need different levels of protection for people's privacy. They also advocate user control over the disclosure of location information. The approach allows for protection of anonymity via 'secret groups'. They argue that in a large, heterogeneous system only the user-agent approach (as opposed to the location service approach) can deliver a meaningful protection of privacy. Xerox PARC's user-centric architecture is spelt out in [14]. Here, the user agent implements the access control decisions as specified by the corresponding user. Access control can also be delegated to a central Location Broker to increase efficiency. Location are not treated as first-class objects in this model, that is, no explicit policy regarding access to a specific location can be specified.

Rizzo and others describe their work on a secure location service for an office environment in [11]. Their location service is constructed of at tree of Locators which are location tracking subsystems. Secrecy is protected by access control to those Locators. Capabilities are employed to allow select access to Locators. (These capabilities could, in principle, be used to specify the range of authorised results for Locator queries.) Organisational policies are expected to be hardwired into the Locators, while discretionary policies can be specified and altered by the individuals who 'own' the location information.

In both cases, the location services have been designed with a concrete implementation-specific security model in mind. There is no general architecture-independent specification of security policy which could by applied to a different location service architecture.

10. Conclusions

In this paper, we have discussed the security requirements faced by locations services deployed in different organisational environments. We have identified two likely deployment scenarios, large organisations and heterogeneous global services.

Both mandatory label-based protection and matrix-based protection can be applied to a location service. In both cases, the traditional approaches need to be generalised in order to be suitable for the location service. This is because location information does not provide an obvious *target object* for policies and labels. If the located-object is treated as the target object, it becomes very hard to specify a access control for all objects at a given location. If the location is made the target object, it is difficult to specify access control for a given located-object. Using both methods in combination is not satisfactory because the access control information would be duplicated. Therefore, we have proposed multi-target policies for discretionary access control, and three-label axioms for mandatory pol-

icies.

Matrix-based access control offers a flexibility and expressiveness far superior to label-based access control. When using a domain-based framework, the access matrix can be specified as a set of canonical policies over groups of objects. Thus, the policy-based approach becomes scalable and manageable. Further, both centralised and decentralised system can use policies. Label-based access control caters for a much narrower set of requirements. Therefore it is only appropriate for use in systems with very specialised requirements.

We have designed and implemented a policy-base security model a location service based on a hierarchy of symbolic locations. Our model allows for flexible protection of organisational and personal privacy. We have identified three levels of protection: access protection, location anonymity and personal anonymity. These protection levels can be provided by either mandatory or discretionary access controls.

In the future, we need to investigate further the application of our approach in a heavily decentralised environment, such as described in scenario I. The challenge here lies in meaningful cross-domain authorisation of access to location information. Especially the possibility of overlapping service areas appears to complicate this problem. Further, we need to improve or rewrite our prototype in order to take into account scalability and performance issues.

Acknowledgements

This work is funded by the UK Engineering and Physics Science and Research Council as part of the MNA programme. We were kindly supported with Illustra software by Informix, Inc. Further, we would like to acknowledge the input of Emil Lupu on security policies.

References

1. S. Castano et al. *Database Security.* Addison-Wesley, 1994.
2. D. Clark and D. Wilson. A comparison of commercial and military computer security policies. In *Proceedings of the IEEE Security and Privacy Symposium*, pages 184-194, 1987.
3. C. Dhawan. *Mobile Computing: A System Integrator's Handbook*. Computer Communications. McGraw-Hill, 1997.
4. Informix, Inc. *Illustra User's Guide*. October 1995.
5. Informix, Inc. *3D Spatial DataBlade Guide*. March 1995.
6. R. Jain and N. Krishnakumar, *Service handoffs and virtual mobility for delivery of personal information services to mobile users*. Technical Memorandum TM-24696, Bell Communications Research, December 1994.
7. B. W. Lampson. Protection. In *Proceedings of the Fifth Annual Princeton Conference on Information Science Systems*, pages 437-443, 1971. Reprinted in Operating Systems Review, Volume 8, Number 1, pages 18-24, 1974.
8. U. Leonhardt and J. Magee. Towards a general location service for mobile environments. In *Proceedings of the Third International Workshop on Services in Distributed and Networked Environments*, pages 43-50, Macau, June 1996. IEEE CS Press.
9. D. Marriott and M. Sloman. Management policy service for distributed systems. In *Proceedings of the Third International Workshop on Services in Distributed and Networked Environments*, pages 2-9, Macau, June 1996. IEEE CS Press.
10. M. Moulet and M.-B. Pautet. *The GSM System for Mobile Communications.* Palaiseau, France, 1992.
11. M. Rizzo, P. Linington, and I. Utting. *Integration of location services in the open distributed office*. Technical Report 10-94, University of Kent, Computing Laboratory, Canterbury, UK, 1994.
12. B. Schilit, N. Adams, and R. Want. Context-Aware Computing Applications. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*. Santa Cruz, December 1994.

13. M. Sloman and K. Twidle. Domains: A framework for structuring management policy. In M. Sloman, editor, *Network and Distributed Systems Management*, pages 433-453. Addison-Wesley, 1994.
14. M. Spreitzer and M. Theimer. Providing location information in a ubiquitous computing environment. In *Proceedings of the 14th ACM Symposium on Operating System Principles*, volume 27 of ACM SIGOPS, pages 270-283, 1993.
15. M. Spreitzer and M. Theimer. Scalable, secure, mobile computing with location information. *Communications of the ACM*, 36(7):27, 1993.
16. M. Spreitzer and M. Theimer. Architectural considerations for scalable, secure, mobile computing with location information. In *Proceedings of the 14th International Conference on Distributed Computing Systems*, pages 29-38, Poznan, Poland, June 1994. IEEE CS Press.
17. K. Twidle. *Domain Services for Distributed Management*. Ph.D. Thesis, Imperial College, Department of Computing, London, UK, May 1993.
18. N. Yialelis. *Domain-Based Security for Distributed Object Systems*. Ph.D. Thesis, Imperial College, Department of Computing, London, August 1996.

The Authors

Ulf Leonhardt is a Ph.D. student at the Department of Computing at Imperial College in London, where he works as a research assistant in the Distributed Software Engineering Group. His research interests include distributed systems, mobile computing, location tracking, and software process modelling. He is a member of the ACM and the SIGMOBILE. Ulf was born in 1970 in Karl-Marx-Stadt (now Chemnitz) in the former GDR. He studied Informatics at the Technical University Dresden, and later Software Engineering at Imperial College. There he gained an M.Eng. degree in 1994.

Jeff Magee is a Reader in the Department of Computing at Imperial College. His research is primarily concerned with the software engineering of concurrent, distributed and mobile computing systems; including analysis tools, languages and support environments for these systems. He studied for his B.Sc. degree in Electrical Engineering at Queens University Belfast and for his M.Sc. and Ph.D. in Computing at Imperial College. He is co-editor of the IEE Proceedings on Software Engineering.