Collaboration and the Knowledge Economy: Issues, Applications, Case Studies
Paul Cunningham and Miriam Cunningham (Eds)
IOS Press, 2008 Amsterdam
ISBN 978–1–58603–924-0

Location-Aware Access Control for Mobile Information Systems

Michael DECKER
Institute AIFB, University of Karlsruhe (TH), Englerstraße 11, 76 128 Karlsruhe, Germany
Tel: +49 721 608-0, Fax: + 49 721 608 5998, Email: decker@aifb.uni-karlsruhe.de

Abstract: Location-Aware Access Control (LAAC) means to consider a mobile device's location when making the decision if a request to perform an operation on a particular resource should be granted or not. This is an approach to tackle specific security-related challenges that come along with the employment of mobile information systems but it can also help to support the interaction between user and mobile device in a context-aware sense. For the realization of LAAC two things are essential: the capability to determine a mobile user's location and an appropriate data model to formulate statements about which operations he or she is allowed to perform on which resources at a particular place. Therefore in our paper we first discuss several locating technologies from the perspective of LAAC. Afterwards we introduce a novel location-aware access control model. One special feature of our model is that it supports dynamic location-constraints, i.e. constraints that can only be determined during the runtime of a mobile application.

1. Introduction

Distributed information systems with portable computers like notebooks or PDAs that may have access to stationary backend systems over wireless data communication are called mobile information systems (MIS). Such systems have a great potential for many application scenarios since they can provide computer support almost "anytime, anywhere".

However portable computers are prone to theft and loss which is occasionally brought to the public's awareness by incidences that hit the headlines, e.g. cases like that reported in [15] when notebooks owned by public institutions with sensitive person-related data get lost. Technical measurements against such mishaps are encrypted storage of data (e.g. encrypted hard disks in notebooks), biometric authentication to unlock mobile computers (most notably readers for fingerprints) or triggering deletion of all data stored on a lost device by sending a special command message (so called "kill pill"). In this paper we discuss location-aware access control (LAAC) as another solution to tackle this problem. Access control is concerned with limiting the access to resources that are managed by information systems, e.g. making the decision which operations (e.g. write, read) a user can perform on a particular object stored in a file system or which services he may use (e.g. printing service, querying database) [1]. It is usually based on the consideration of the user's identity as well as individual permissions assigned to the resources under control. LAAC also evaluates the user's current location. So policies like "a nurse is only allowed to access the electronic health record using a PDA while staying in the ward" can be enforced.

But LAAC is also beneficial for reasons beyond security aspects: due to their size mobile computers dispose only over relatively constrained means for user interaction; the display is tiny and of limited quality (contrast, resolution, and colour depth) and data input is cumbersome because there is no full keyboard. One idea to mitigate this is called "context-awareness" [7] and comprehends the evaluation of information about the user's current situation to support his interaction by reducing the number of interaction steps

required to perform a particular task. LAAC can be "misused" to hide buttons, options and data sets on the graphical user interface that are not needed at particular locations and thus can help to reduce the data that has to be presented on the display and the navigations steps. Imposing location-aware access restrictions can be motivated by several considerations:

- It may not be plausible to access particular resources outside certain location, e.g. a travelling salesman doesn't need access to a customer record if he stays outside the respective sales district.
- At some places it might be deemed as to dangerous to have access to confidential resources, e.g. at frequented public places, outside a company's premises or in countries where espionage has to be feared.
- Some functions of a MIS may be only useful in the proximity of certain facilities, e.g. remote control of devices installed in rooms like machines, multimedia-equipment or air-condition.
- Staying at particular locations constitutes a form of authentication if physical access to that location is secured, e.g. an area, building or part of a building secured by human guards or doors. In this case LAAC may obviate the need for identity-based authentication and thus providing anonymous access to MIS.

There are already some simple forms of LAAC that can be found in practice: movies on DVD can be protected by the so called region code so discs can be produced that can only be viewed with hardware players manufactured for the European market. There are a few personal navigation devices on the market that are protected by a password which can be only reprogrammed at the place where the initial password was set. Payments with credit cards may be rejected if the payment is attempted at a place where the card holder usually doesn't stay, e.g. shopping in a foreign country or online payment using an internet terminal with an IP-address originating from a foreign country. In some countries there are pay-TV decoders which query the decryption key over a telephone line; the provider will only sent the key if the telephone connection lies in the respective country.

Two things are essential for the realization of advanced forms of LAAC: the possibility to locate a mobile device and an appropriate data model to formulate access restrictions with regard to locations; such data models are called access control models (ACM). In our paper we want to introduce a new location-aware ACM (LAACM) which builds upon the metaphor of different classes of documents. One special feature of our model is the ability to formulate dynamic location-constraints, i.e. location-constraints that can only be determined during the runtime of a process but not in advance at design time. This feature isn't supported by any LAACM we are aware of. We will also discuss several techniques for locating with respect to their suitability for LAAC. Again this aspect is neglected in the pertinent literature.

The remainder of our paper is structured as follows: in section two we first cover the basic models for access control and then discuss the most important LAACMs. Section three is dedicated to locating techniques for LAAC. In section four we introduce an LAACM along with a location-model whereby the focus lies on the possibility to formulate dynamic location constraints. Finally the paper ends with a conclusion given in section five.

2. Models for Access Control

2.1 Basic Models

Access control is the process of determining if a subject's request to perform a particular operation on an information system's resource should be granted or not. Subjects are human users or programs acting on behalf of a user; resources are files, database objects or services. The software component that is responsible for the decision if a request should be allowed or not is termed "reference monitor". There are three basic types of access controls

[1]: discretionary access control (DAC), mandatory access control (MAC) and role-based access control (RBAC). DAC is the approach used by most contemporary operation systems: a resource has an owner who has all rights which he can grant to other subjects; the access decision is made based on the identity of the subject and his rights on that object. For MAC the reference monitor evaluates system-wide rules to make a decision according to labels assigned to subjects as well as objects. These labels might denote several security levels and the rules enforce policies like "a subject with clearance confidential is not allowed to read an object with classification top secret". MAC originally was developed for military systems but meanwhile there are also implementations for civil usage available, e.g. SELinux or AppArmor. RBAC's basic notion is it to have roles that represent job descriptions within organizations, e.g. "manager", "secretary" or "accountant".

These roles are collections of permissions that are assigned to subjects whereas a permission is the right to perform a particular operation on a particular object. It is not allowed to assign permissions directly to a user. The great advantage of RBAC is that it saves a lot of administrative work: if a subject's job within an organization changes (e.g. employee hired or promoted) the administrator just has to adjust the roles assigned to this subject and not to assign all the individual permissions.

2.2 Models for Location-Aware Access Control

In literature several proposals for LAACM can be found; most of them are variants of RBAC. The basic idea to obtain a location-aware RBAC model is to switch off certain RBAC entities if the user stays outside predefined locations. For example in the GEO-RBAC model [5] roles can be disabled depending on the user's location. In the SRBAC-Model [8] the assignment of permissions to a role is active only at defined locations. In LoT-RBAC [16] location-restrictions can be bound to the user-role-assignment, to roles alone and to the role-permission assignment. There are similar LAACMs (e.g. GRBAC, DRBAC, LRBAC) but due to space limitations we can't discuss them here.

Ray and Kumar [13] developed a location-aware model for MAC. In their model security levels are assigned to individual locations. A user is then not allowed to use the MIS if his clearance is below a location's classification and resources can only be accessed at locations that have a classification at least as high as the resource. Leonhardt and Magee [10] propose a location-aware DAC model to formulate policies that express under which conditions a user is allowed to learn about another user's location.

3. Technologies for Location Determination

LAAC relies on the ability to determine a mobile device's location, of course. There are many methods to locate a mobile device, see [9] for an overview. Two generic approaches for locating can be distinguished (see also figure 1): in the case of self-locating the mobile device calculates its position based on signals emitted by a locating network (e.g. satellites, base stations or beacons); in the case of remote-locating the locating networks receives signals from the mobile device and calculates its position. Depending on the deployment scenario it may be necessary that for self-locating the mobile devices forwards its location fix to a stationary backend system; for remote locating the network has to report the location either to the mobile device or to a backend system.

If LAAC is employed not merely to realize interaction support but rather for security reasons we have to assess how prone to manipulations particular locating technologies are. The term "spoofing" with regard to locating is used in literature to describe manipulations performed by the owner of the mobile device as well as by external third-parties. To differentiate we call the former "internal spoofing" (case 2 and 3 in figure 1) and the latter "external spoofing" (case 1 and 4 in figure 1). External spoofing attacks are based on

manipulations of the signals for locating exchanged between locating network and mobile device. For GPS (self-locating) an external attack would be to broadcast fake satellite signals that superimpose the genuine signals [16]. In the case of remote-locating external spoofing would mean to fake the signals that are sent by the mobile device to the locating network. However both attacks can be prevented if the respective signals carry a digital signature which is indeed planned for the EU's Galileo system.

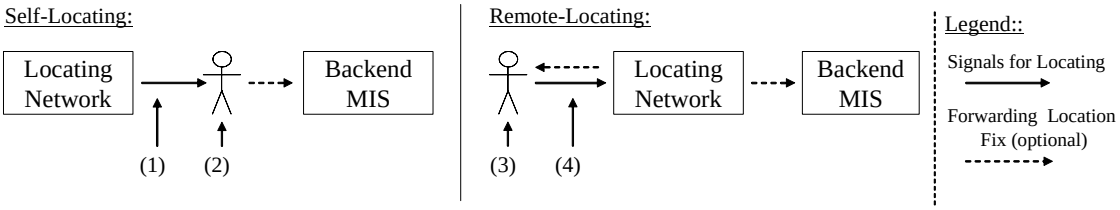

Figure 1: Different Points to Manipulate Location Determination

The case of self-spoofing however is more interesting and is therefore discussed here in more detail: In the case of self-locating the user calculates his own location fix and forwards it to the backend system. He could therefore just "invent" an arbitrary location. One suggesting approach to prevent this attack is that the mobile device has to forward some piece of information that is only visible at the alleged location. This is the basic concept of the CyberLocator system that supplements GPS [6]: here the mobile device forwards also the raw signals received by the GPS satellites. Since the raw signals are influenced by atmospheric effects they cannot be predicted. The backend then compares these signals with signals reported by trusted reference stations not farer away than 2.000 miles. However it is thinkable that the mobile user stays outside the designated location and forwards signals he receives from a colluding party that stays inside the location; this is called "rerouting attack". To prevent rerouting attacks the system demands that the mobile users forwards the signal with a delay of not more than 5 ms since the latency for the rerouting transmission would exceed this time span. Another idea is to broadcast a special location key that can only be received in a particular area [4].

There are other approaches where the locating network sends a special message (including a non-predictable random number) to the mobile device that must be returned within a certain time span, e.g. [14]. This time span is calculated based on estimation of the signals travel time for the distance between mobile device and base station of the locating network. If the mobile device isn't at the location it claims to be its answer won't reach the locating network in time because radio signals cannot travel faster than the speed of light.

There are also proposals for spoof-resistant locating when the reference monitor is operated on the mobile device and no permanent data communication is available. The system for location-aware digital rights management described in [11] is based on satellite navigation and a tamper-proof hardware module as reference monitor. The hardware module hosts a precise clock, an unit for location-determination based on the satellite signals and a decoder to decrypt the content under protection. It is assumed that the satellite signals carry a timestamp and are digitally signed. So the locating unit can determine if navigation signals are manipulated or replayed; it his however necessary to synchronize the clock every couple of hours.

4. Document-Centric Access Control Model with Dynamic Location-Constraints

4.1 Core Model

Our model builds upon the metaphor of virtual documents that are attached to particular places. Each document is an instance of exactly one document class and can be accessed by

a user within a defined range. For this paper we think of these documents as files to be edited by some kind of text editor. However the ACM could be easily extended for more general applications where a document instance is a container for different data objects that are processed by several applications. The rationales behind having several document classes is that it allows us to assign different sets of default permissions appropriate for different application scenarios, e.g. class “ordinary note” has the default to allow write operations to users with role “employee” but for class “manager note” only operation “write” is assigned to this role. Further the rules for the dynamic creation of location restrictions will vary greatly depending on the respective application scenario. The user can also configure his device so that only documents of selected classes are displayed.

We depict the model as UML class diagram in figure 2: in the upper part the location model can be found whereas in the lower part the core model is depicted. There are four abstract classes in the model whose names start with “abstract”: an abstract class cannot be instantiated but subsumes properties that are shared by several non-abstract classes that inherit from the abstract class.

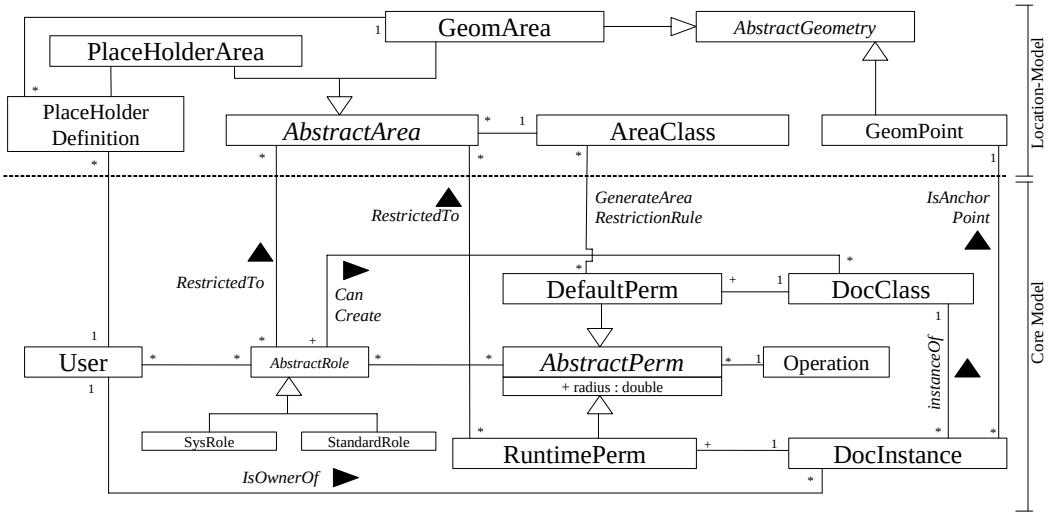

Figure 2: Data Model (Cardinalities for Associations: “” stands for “0..N”, “+” for “1..N”)*

Following the main idea of RBAC we employ the concept of roles: users can be assigned to roles and roles get assigned to permissions. The permissions assigned to roles are actually abstract permissions. Permissions make a statement about an operation the users of a role are allowed to perform on a document. Operations for the document metaphor are “read”, “write”, “append”, “delete” or “alterPermissions”. Class RuntimePerm inherits from AbstractPerm; each instance of this class represents the assignment of one operation to one document instance for one role. Like the name implies these permissions are evaluated to decide if a user is allowed to perform a particular operation on a particular document instance. For example the triple “clerk;customerNote#123;read” allows all users with role “clerk” to perform the operation “read” on document instance “customerNote#123”. If a new document instance is created the DefaultPerm*s* are evaluated to find out which initial RuntimePermission*s* have to be set for the respective DocumentClass. Since the operation “alterPermissions” may allow users to alter the permissions assigned to a document instance it is necessary to have a copy for the permission set for each document instance.

The roles used in the model are either instances of class SysRole or StandardRole since their common super class AbstractRole cannot be instantiated. SysRoles are system roles that cannot be deleted and must be present for each ACM installation; two such roles are “owner” and “admin”. StandardRoles however are application-specific roles that must be created for each ACM installation according to the respective scenario and organization.

Using the association “isOwnerOf” we store the owner who created a document instance. The system role “owner” enables to assign special permissions for a document instance’s owner. Another association is between AbstractRole and DocClass and named CanCreate. Using this association we can express which roles are allowed to create instances of a particular document class.

4.2 Location Model

At creation of a new document instance the current location of the creating user is assigned as anchor point to the document instance; this anchor point is represented by an instance of class GeomPoint. There is also an attribute called “radius” in class AbstractPerm. If this attribute is set to a non-negative value it says the respective permission can only be used if the user is currently not farther away from the document’s anchor point than specified by the radius value. This allows us to have individual location restrictions for each permission and each role. For example a user with role “boss” can read documents of a given class at a distance of 100 meters and write to them at a distance of 10 meters but role “employee” has no write permissions and can only read documents at a distance of up to 5 meters.

Instances of classes AbstractRole and RuntimePerm can be associated with class AbstractArea; if this is the case the respective instance can only be activated while the respective user stays within the denoted area. So we can have a role “nurse” that can only be activated while the users stay within the premises of the hospital. We could also think of other points in the model where a location restriction could get attached to, e.g. to users (i.e. a user is only allowed to access the MIS while staying in a particular area), the canCreate-association (the respective role is only allowed to instantiate that document class in certain areas) or the role-permissions association (a role can only active a permission in certain areas). However due to space limitation we cannot consider this here.

Each AbstractArea stands in association with exactly one AreaClass. An AreaClass is just a symbol to classify AbstractArea-instances into human-understandable categories like “cities”, “buildings”, “countries” or “regions”. AbstractArea has two subclasses: GeomArea and PlaceHolderArea. Instances of GeomArea denote geographic areas; for implementation purposes we assume that these areas are non-empty polygons. The other subclass PlaceHolderArea represents symbolic names of personal areas, i.e. areas that will differ from user to user. Examples are “home town”, “office room” or “sales district”. To resolve such a PlaceHolderArea to obtain a GeomArea the reference monitor has to resort to the PlaceHolderDefinition which assigns a GeomArea for pairs of “user” and “PlaceHolderArea”, e.g. for user Alice the PlaceHolderArea “home town” could point to the GeometricArea “Stockholm”.

RuntimePerms can be restricted to AbstractAreas, so the respective role is only authorized to perform that permission in certain areas. If a permission is location restricted the respective DefaultPerm is associated with an instance of AreaClass. At creation time of the respective document class the reference monitor checks for a GeomArea-instance that is an instance of the respective class and contains the current position of the user. If there is such a GeomArea it will be assigned to the newly created RuntimePerm.

A novel feature of our ACM are dynamic location restrictions, i.e. the geometric areas that restrict the respective permission are determined during runtime of the model and don’t have to be defined in advance by the administrator like for other LAACMs. There are three different kinds of dynamic location restrictions: (1) restricting permissions according to the anchor point where the document was created and the radius; (2) PlaceHolderAreas are also dynamic location restrictions because to obtain the GeomArea the reference monitor has to know which user is trying to play the respective role; (3) the evaluation of the AreaClass associated with a DefaultPerm to obtain an AbstractArea instance as location restriction for a RuntimePerm instance. To obtain these dynamic location restrictions there are several

associations between classes of the core and the location model. This degree of integration between core and location model cannot be found in other LAACMs (if they comprise an explicit location model at all).

4.3 Scenarios

To exemplify the employment of our model we sketch some example applications that use the document metaphor in a natural manner. In all mentioned applications the mobile devices accesses a service on a stationary backend system and the reference monitor resides also on the stationary backend. In literature some descriptions of systems that are based on the metaphor of location-aware documents can be found, e.g. [2] and [12]. However these systems are not based on a general data model or even a LAACM.

Using the "Virtual Graffiti" service a user with role "graffiti-author" can deposit messages at his current location, e.g. "do not visit this pub, the beer is horrible". This message can be read by users with role "graffiti-reader" while staying not further away than 50 meters from the respective anchor point. The message can be altered and deleted by the owner from a distance of up to 1 km; the administrators of the system can do this from any location. If we grant writing permissions also to the "reader"-role we obtain a service where a document can be edited by a community of people; the documents of this service would be some kind of location-aware wiki-pages. If we restrict the "reader"-role to PlaceHolderArea "WikiServiceArea" each user can define individual areas where he would like to use this service. There could also be a document class denoted "personal reminder" whose instances can only be altered, read and deleted by the creating user.

But using this model we can also implement services for mobile workers: service technicians sent to facilities for maintenance work could deposit notes to write down what has to be repaired; they can only alter these notes while staying near the respective facility. For a travelling salesman we could have a service where each document instance represents a customer record. He is only allowed to alter the record (e.g. to enter orders) while staying at the premises of the respective customer. Back office workers however are allowed to alter the record only while staying at the PlaceHolderArea "personal work office", but there is no anchor-point restriction imposed to these permissions.

The next service is an indoor scenario: for a mobile healthcare application a patient's medical record is represented by a document instance. The role nurse is only allowed to view or edit a health record while staying on the hospital's premises, however role "physician" has also the PlaceHolderArea "private residence" as location restriction so physicians can work at home. Abstracting from the document metaphor we could also have distinctive parts of a health record (e.g. section for "vital parameters" and section for "doctor's order" or "diagnosis") which are protected by individual permissions (e.g. "write permission" only for "doctor's order" section). If there is an *AreaClass* "hospital ward" we could also restrict selected permissions (e.g. "append vital data section") to the ward where the patient has his bed.

5. Conclusion

The main contribution of the paper at hand was the description of a location-aware access control model that follows the metaphor of virtual documents. One novel feature of our model are dynamic location constraints, i.e. it is able to express location constraints that can be only determined during runtime of an application. Several application scenarios for consumers as well business users were sketched to demonstrate the versatility of the model.

One notion for further elaboration of the model are security labels that can be assigned to different areas: so a document created at a location classified as "secret" can only be read

and edited while staying at a location classified at least as “secret” or above. Further it is worthwhile to devise how to deal with imprecise location determination.

Acknowledgements

This work has been funded by the Federal Ministry of Economics and Technology of Germany (BMWi, Contract No. 01MD06012, Project “MODIFRAME”). The responsibility for the content of this paper lies solely with the author.

References

[1] M. Benantar, Access Control Systems. Springer, Heidelberg et al., 2006.
[2] P.J. Brown, The Stick-E Document: A Framework for Creating Context-Aware Application. Electronic Publishing, 8(2 & 3), pp. 259-272, 1995.
[3] S.M. Chandran & J.B.D. Joshi, LoT-RBAC: A Location and Time-Based RBAC Model. Proceedings of the 6th International Conference on Web Information Systems Engineering (WISE ’05), pp. 361-375, 2005.
[4] Y. Cho, L. Bao & M.T. Goodrich, LAAC: A Location-Aware Access Control Protocol. Third Annunal International Conference on Mobile & Ubiquitous Systems, pp. 1-7, 2006.
[5] M.L. Damiani & E. Bertino, Data Security in Location-Aware Applications: An Approach Based on RBAC. International Journal of Information & Computer Security, 1(1/2), pp. 5- 38, 2007.
[6] D.E. Denning & P.F. MacDoran, Location-Based Authentication: Grounding Cyberspace for Better Security. Computer Fraud & Security, pp. 12-16, February 1996.
[7] Dourish, P.: What We Talk About When We Talk About Context. Personal Ubiquitous Computing, 8(1), 2004, 19-30.
[8] F. Hansen & V. Oleshchuk, SRBAC: A Spatial Role-Based Access Control Model for Mobile Systems. Proceedings of Nordsec ’03. pp. 129-141, 2003.
[9] J. Hightower & G. Borriello, Location Systems for Ubiquitous Computing. IEEE Computer Magazine, 34(8), pp. 57-66, 2001.
[10] U. Leonhardt & J. Magee, Security Considerations for a Distributed Location Service. Journal of Networks and Systems, 6(1), pp. 51-70, 1998.
[11] T. Mundt, Location Dependent Digital Rights Management. Symposium on Computers and Communications (ISCC), pp. 617-622, 2005.
[12] P. Persson et al., GeoNotes: A Location-Based Information System for Public Spaces. Designing Information Spaces: The Social Navigation Approach, Springer, pp. 151-173, 2002.
[13] I. Ray & M. Kumar, Towards a Location-based Mandatory Access Control Model. Computers & Security, 25(1), pp. 36-44, 2006.
[14] N. Sastry, U. Shankar & D. Wagner, Secure Verification of Location Claims, Second Workshop on Wireless Security, ACM, pp. 1-10, 2003.
[15] The Herald: Personnel May Sue MoD Over Stolen Laptop Data (8th February 2008), http://www.theherald.co.uk/misc/print.php?artid=2028852
[16] J.S. Warner & R.G. Johnston, GPS-Spoofing Counter Measurements. Los Alamos National Laboratory, Technical Report, 2003.