IADIS International Journal on Computer Science and Information Systems
Vol. 5, No.1, pp. 26-44
ISSN: 1646-3692

LOCATION-AWARE ACCESS CONTROL: AN OVERVIEW

Michael Decker. *Institute AIFB, Karlsruhe Institute of Technology (KIT), Kaiserstr. 89, 76 128 Karlsruhe, Germany*

ABSTRACT

The basic notion of *Location-Aware Access Control (LAAC)* is to evaluate the current position of a mobile user as provided by a locating system like GPS when making the decision if a user's request to perform a particular operation on a particular resource under the control of an information system should be granted or denied. LAAC is a mean to forbid the access to computer resources when the mobile user stays at a place where it is not reasonable or not safe enough to access the respective resources. For example, using this approach a policy could be enforced that demands that a confidential document (resource) can only be read (operation) while staying on the premises of a particular company. The aim of this paper is to give an overview on works in the field of LAAC. The special focus is on *Access Control Models (ACM)* which are the data models needed to formulate and maintain location-aware access control policies.

KEYWORDS

Mobile Information Systems, Access Control Models, Location-based Services, Mobile Security

1. INTRODUCTION

Access Control is the process of determining if a given request made to an information system should be granted or not. The request originates from a user (termed "subject" in the pertinent parlance) and aims to perform a particular operation (e.g. read, write, execute, create) on a particular resource like a data object (e.g. electronic document in file system, database object) or service (Benantar, 2006). With the advent of *Location-based Services (LBS)* for mobile devices like PDAs, smartphones or notebooks the idea was developed to evaluate the location of a user respective his mobile device as further or even the *only* input for an access control decision; this is called *Location-Aware Access Control (LAAC)*. It is the aim of this article to give an overview about the most important works by other authors on the field of LAAC.

As example for LAAC we regard a travelling salesman who is equipped with a PDA that has a GPS-receiver. If he wants to access a document about a customer using his PDA the access should only be granted when he stays in the city where the respective customer has his premises. But there are many more scenarios to motivate the employment of LAAC:

- Healthcare professionals in a hospital should only be allowed to make entries to a patient's electronic health record using a PDA if they stay in the room where the patient has his bed. This way mix-ups of patients' records can be prevented and if a PDA is lost or used at public places privacy infringements are avoided.
- If a mobile device is used as remote control for home entertainment equipment (TV kit, audio system, video projector), housing technology (light, air conditioning system, window shades), portable or stationary machines in factories or door locks it would be reasonable to enable these functions only when the mobile user is in close proximity of the respective controllable device.
- The access for some digital services could be restricted to certain areas, e.g., multimedia contents or e-books that are only licensed for a particular country. Restricting a service to a particular area is especially of interest when this service is provided free of charge for certain user groups, e.g., wireless web access or access to information services that should be only available for current customers of enterprises like stores, hotels, cafés or amusement. We can also think of literal campus licenses for education institutions where the fee to pay depends on the size of the area where the customer wants to use the software or to access the digital content (e.g., electronic text book). Several articles dealing with locating-technologies secured against "spoofing" (e.g., Mundt, 2006) are motivated by scenarios stemming from "location-aware *Digital Right Management (DRM)*", e.g., a consumer buys a license to playback multimedia content on his mobile device only within one country; or the provider of set top boxes for the decipherment of television programs wants to prevent a consumer from operating his box outside his apartment, so that the box cannot be misused for entertainment at public places.
- An enterprise with employees traveling to countries all over the world might wish to forbid the access to some confidential document (e.g., research reports, pending patents, and business figures) in a country where industrial espionage has to be feared, competitors operate or the legal system is not trustworthy. In some countries the use or export of encryption software is restricted so that a service provider might want to employ LAAC to disable these functions.
- An organization can employ LAAC to circumvent that its employees use their mobile devices for communication while they stay at the organization's premises where conventional (and much cheaper but not so convenient) means of communication (e.g. fixed-line telephone) are available.
- A multi-national company offering products for consumers may want to guarantee that person related customer data is accessed in any country except the country where the data was acquired. Meanwhile some countries have even laws which demand this.

LAAC is a mean to tackle specific security issues that come along with the employment of mobile devices: such devices get easily lost or stolen or are used at public places where unauthorized individuals could sneak a peek over the user's shoulder (termed also "shoulder sniffing", "shoulder surfing" or "over-the-shoulder-attack"); further, wireless data transmission could be eavesdropped. If access to a computer system is restricted to particular locations the consequences of such mishaps are mellowed or avoided at all. But the usefulness of LAAC is not limited to security aspects; this concept can also be applied to improve the usability of mobile applications: referring to the traveling salesman scenario given above the LAAC could be used to hide all the documents on the PDA screen that are not relevant for the mobile user at his current location. This way the interaction between human user and mobile device is supported since the number of data items to be displayed on the tiny screen is reduced; furthermore, it also reduces the number of navigation steps required by the user because irrelevant data items don't have to be skipped. Since data input on a mobile computer is cumbersome this greatly improves the usability of a mobile application.

While most research contributions consider location-awareness as additional component for an access control mechanism it is also thinkable to employ location information as the only input for an access control decision and thus go without the need to determine the user's identity. This might be a preferable property if knowledge about who made a particular request to an information system can lead to an infringement of privacy. For example, Alice's employer could inference that she never comes to work before 10 a.m. if Alice never makes access attempts to the company's file server before that time. For access decisions without regarding the user's identity it is necessary that staying at a particular place is sufficient for certain privileges, e.g. if the enterprise campus is secured by walls and guards or if a service should be available to each customer currently staying in a restaurant or theme park.

Is LAAC a form of LBS? From a high-level conceptual view it is since LBS are services that evaluate a user's location to adapt their behavior accordingly (Küpper, 2007). This is exactly what LAAC does, however, LBS usually evaluate the location to provide more comfort to the user, e.g., by showing the nearest *Point-of-Interests (POIs)* without prompting the user for his position (which he maybe doesn't know because he lost his way). So from a *Human-Computer-Interaction (HCI)* perspective LAAC is not an LBS since it is usually not a mean for supporting the user but rather to "vex" him by denying access to resources he might wish to access. In mobile computing LBS are generalized to context-aware services, i.e., further information (e.g., time, profile information, available resources, nearby people) is gathered at runtime and evaluated to dynamically adapt the service.

The remainder of the article at hand is organized as follows: in section 2 we cover the necessary basics concerning access control. Section 3 is devoted to describe various *Location-Aware Access Control Models (LAACM)* that can be found in literature; this description is structured according to the basic approaches of conventional ACMs, namely *Discretionary Access Control (DAC)*, *Mandatory Access Control (MAC)* and *Role-Based Access Control (RBAC)*. Different ideas how to prevent or detect the manipulation of locating technologies are the topic of section 4. There are already a few forms of LAAC that made their way into commercial products and which are mentioned in section 5. In section 6 some issues for future research are discussed before we conclude in section 7.

2. ACCESS CONTROL

There are three basic approaches for access control (Benentar, 2006; Ferraiolo, Kuhn & Chandramouli, 2007): *Discretionary Access Control (DAC)*, *Mandatory Access Control (MAC)* and *Role-Based Access Control (RBAC)*. Many research publications describe these three in a way that suggests that they are distinct classes in the mathematical sense; however, since RBAC can be configured to act as DAC or MAC, this is not the case. The data model behind an access control approach is called *Access Control Model (ACM).*

Most readers will be familiar with DAC even if they never heard about it since this approach is implemented by most contemporary operation systems: here the user who created a resource (e.g. a image file) is the owner of that resource and therefore has all permissions on that object. However, it is at his *discretion* to grant individual permissions to other users, e.g., if Alice created a text document she can grant the right to read this document to Bob and grant to Claire the right to perform the operations “read” and “write” on that document. A DAC system even might allow that Bob or Claire grant their rights to further users. A natural way to write down this model is the *Access Control Matrix* which represents a simple form of an ACM (Lampson, 1974): in this matrix each row stands for one user and each column for one resource so each element in the matrix represents one combination of user and object. Each element contains the permissions that the respective user is allowed to perform on the respective resource. However, since this matrix will have many empty elements (i.e. it is sparse matrix) for implementation purposes other data structures are preferable: e.g., many file systems assign an *Access Control List (ACL)* to each file (resource) where each entry defines a particular permission for one user; so the ACL represents the information of one column in the matrix. Another approach is to take the information of a single row to obtain so called *Capabilities* and assign these *Capabilities Lists* to the respective user.

MAC comes from the military domain. For this approach there is an ordered list of security levels (Bell, 2005; Benantar, 2006), e.g., “Top Secret” (TS, strictest), “Secret” (S), “Confidential” (C) and “Public” (P). Then each resource gets one of these security levels (called “classification”). Further, each subject gets assigned to one security level (called “clearance”). Based on this and a set of rules the information system itself can decide which accesses are allowed and which not. One common rule is the “no-read-up” rule that says that users are only allowed to read resources that are classified not higher than themselves, e.g., a user with a clearance of “Secret” can read documents with a classification of “Secret” or “Confidential” but not “Top Secret” documents. This form of access control is termed “mandatory” since every resource is controlled without asking the user for an explicit configuration; that’s why MAC is sometimes also called “system-based Access Control”. Today MAC is not only employed in information systems for military intelligence but can also be found in civil software products to provide protection against misconfigurations by users or flawed software components, e.g. *SELinux (Security Enhanced Linux)* or the feature *“Label Based Access Control” (LBAC)* of the DBMS *“DB2”* by IBM.

RBAC (Ferraiolo, Kuhn & Chandramouli, 2007) is based on the observation that in most organizations the different job descriptions are quite stable, while employees change their jobs often (e.g., entering/leaving the organization, promotion for higher position, holiday replacement). So RBAC is based on the concept of a “role”: each role represents a distinct job in the organization (e.g. secretary, manager, developer, or trainee) and is assigned to the permissions that are necessary to perform that job. A permission is usually interpreted as the right to perform a particular operation on a particular object. The actual users are assigned to the respective role. If a new employee enters the organization, it isn’t necessary to assign a lot of individual permissions to him but it suffices to assign a few roles to him. The roles act as mediators between users and permissions. It is forbidden to assign permissions directly to users. Further, it is possible to define an inheritance relationship between roles, e.g., a role “senior consultant” might inherit all the permissions that are assigned to role “junior consultant”. So senior consultants have at least the permissions that junior consultants have.

Another feature of RBAC is "Separation of Duties" (SoD; Sandhu, 1990) that is available in two forms: with "Static Separation of Duties" (SSoD) subsets of roles can be defined that cannot be assigned to one user at the same time. An example for roles that should be mutual exclusive are "cashier" and "financial inspector" — if a cashier could also act as his own financial inspector he easily could obscure faulty or fraudulent transactions he made. The second form of SoD is "Dynamic Separation of Duties" (DSoD): here subsets of roles are defined that cannot be activated within the same session. If we regard a workflow instance as a session we could use DSoD to mark the roles "paper author" and "reviewer" for a conference management system as mutual exclusive; this way a reviewer is still allowed to submit a paper but he cannot "review" his own paper.

To give an impression how these access control models are related to other security concepts we depict the "access control stack" in figure 1: On the uppermost layer we have security policies: these are documents written in natural languages (e.g., English) to express what is considered as "security" by an organization. The next layer represents ACMs which act as formal models to write down what is said in the policies; the article at hand focuses on this layer. To actual enforce the models we need the next layer which we call "technical measures": these are implementation details or technical components. One important component for access control is the "reference monitor" that intercepts each request made to a resource and possibly prohibits it. Another technical measure is encryption of data; there is even a special algorithm for location-dependent symmetric encryption (Liao & Chao, 2008) where the coordinates of the user's location are one input parameter for the generation of the secret key. Further technical measures are network firewalls, biometric devices to determine a user's identity or tamper-proof hardware modules.

It is quiet common to combine different approaches, e.g., to employ MAC and DAC together, so that MAC can intercept errors a human user might have in his DAC configuration or to harden a software system for the case of flawed components. ACMs are just one kind of so called security models; other kinds of security models are *Inference Control Models* or *Data Flow Models*. Since ACMs are the most prominent kind of security models some authors use the term "security model" as synonym for "ACM."

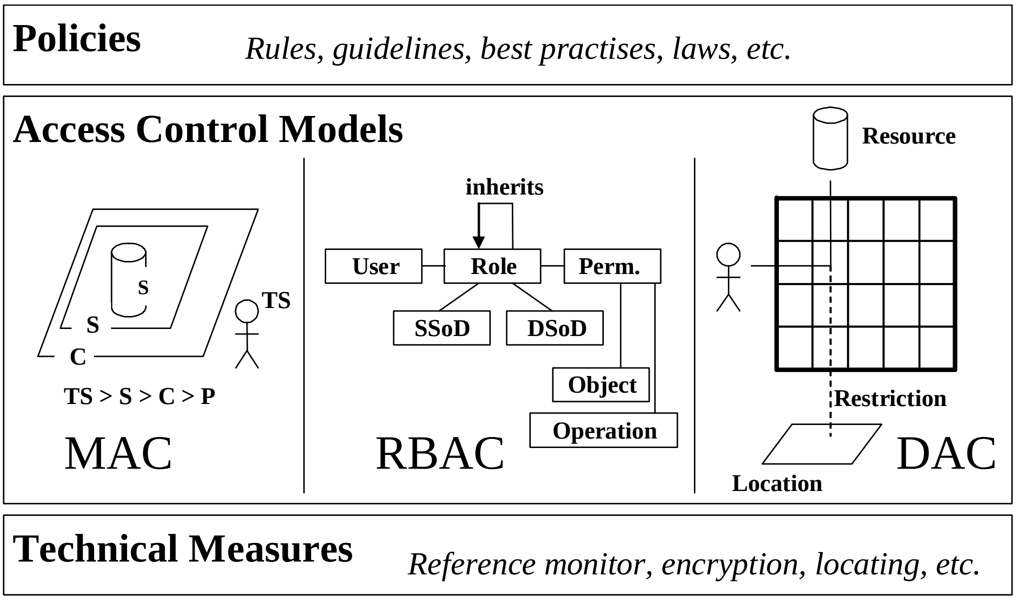

Figure 1. Access Control Stack

3. LOCATION-AWARE ACCESS CONTROL MODELS

In this section we describe several ACMs that are location-aware. The description is structured according to the three basic approaches for ACMs, namely RBAC (section 3.1), DAC (section 3.2) and MAC (section 3.3). Two further subsections cover process-aware LAACMs for workflow management systems (section 3.4) and LAACMs for database management systems (section 3.5).

While surveying the pertinent publications we had the impression that the majority of works in the area of LAACMs stems from the scientific community interested in Access Control and not from the community dealing with Location-based Services. It is also interesting that the majority of LAACMs are extensions of RBAC so it is hard to find a location-aware variant of DAC or MAC. The reason for this might be that RBAC as the most modern approach for ACMs enjoys greater interest in the pertinent scientific community than the older approaches DAC and MAC.

3.1 Location-aware RBAC

Hansen & Oleshchuk (2003) propose an extension of RBAC called *“Spatial RBAC” (SRBAC)*. As deployment-scenario the model assumes a cellular network (e.g., WLAN). In this model the assignment of roles to permissions is location-aware, i.e. if the user is outside a particular region (defined as the area covered by one or more cells of the wireless network) individual permissions for a given role can be “switched off”. For role inheritance there are two possibilities: either all location-restrictions are inherited as well or new location-restrictions can be defined. Also separation of duties is made location-aware: it can be formulated that a user is not allowed to activate a particular pair of roles at the same location.

Damiani et al. (2007) describe *GEO-RBAC* (see also Bhatti et al., 2008). This model employs the location model of the *Geographic Markup Language (GML)* which is based on the notion of features: features are objects that have a spatial extent and are an instance of one feature type. The roles in this model are called "spatial roles" because they have a role-extent; if the user is not inside that role-extent the role is disabled. Another prominent property of GEO-RBAC is the distinction of *role schemas* and *role instances*. Each instance belongs to exactly one role schema. Permissions can be assigned to both role schemas and role instances. Role schemas define a feature type for the role extent whereas role instances define a feature. There are also two separate inheritance hierarchies: one for role schemas and one for role instances. Damiani et al. proposed also an XML-based language to exchange instances of GEO-RBAC models.

Another LAACM is "LoT-RBAC" by Chandran & Joshi (2005). "LoT" stands for "Location and Time", so this model is also able to express temporal constraints, e.g., that a given role can only be activated on working days from 9 a.m. to 5 p.m. o'clock. The model incorporates a simple location-model that distinguishes between logical and physical locations: Logical locations are classes of physical location, e.g., for "city" as logical location there might be "London" and "Amsterdam" as physical locations or instances. Location- and time-restrictions can be assigned to three different components in the model: to the association between user and roles, to the roles itself (like in GEO-RBAC), and to the association between roles and permissions (like in SRBAC). A location restriction for the association between user and roles could be used to enable Alice's role "secretary" only when she stays on the premises of her company. The authors of LoT-RBAC also describe a formalism for the description of "triggers" to activate and deactivate components in the model depending on changes of the user's spatial-temporal context.

LRBAC (Location-Aware RBAC) is a model developed by Ray, Kumar & Yu (2006). It allows restricting the activation of roles to particular locations. A more prominent feature is that it is also possible to restrict the assignment of roles to users to particular locations. To motivate this feature the example of a conference attendee is given that has to be at the location "registration desk" to obtain the role "conference visitor"; the role "citizen of country X" could also be only attained if the respective user currently resides within the territory of that country. Further, the model allows assigning location-restrictions to objects, so there could be a research report that can only be accessed form within one country.

Another model that not only considers location but also time as context to formulate constraints is *STRBAC* by Ray & Toahchoodee (2007), whereas the "ST" stands for "spatio-temporal". This model allows making roles and the role-permission-assignment location-sensitive. The prominent feature of this model is that it provides different forms of inheritance and SoD with regard to time and/or location. For inheritance it provides four modes: "unrestricted inheritance" means that location and time constraints are not inherited at all; for "time restricted inheritance" and "location restricted inheritance" time respective location constraints are inherited. Finally, for "time location restricted inheritance" both location and time constraints are inherited. For static SoD (SSoD) there are also four forms: in the "weak form" two roles related by SSoD cannot be assigned to the same user at the same time and for the same location. The "strong temporal form" of SSoD means that if a user was assigned to a role *x* he may not get assigned to another role *y* at the same location at any time; the "strong spatial form" means that two roles are mutual exclusive at the same time at any location. Finally, the "strong form" of SoD means that two or more given roles cannot be assigned to the user for any time at any location. For dynamic SoD there are also four forms but they refer to the roles a user can activate for a single session.

Finally, there are other context-aware RBAC-variants that don't focus on location as only or main context parameter: TRBAC is the time-aware RBAC and designed to regard time (e.g., recurring intervals like working hours) for access control decisions (Bertino et al., 2000). Moyer & Ahamad (2001) describe *GRBAC (Generalized RBAC)*, a model that introduces environment roles which are used to "[…] capture security-relevant information about the environment for use in GRBAC policies […]". As examples for such context information time, weather or CPU/network load are mentioned. Context roles are only activated if the defined environment situation is currently met.

3.2 Location-aware Discretionary Access Control

Since DAC is the ACM approach most employed in the "real world" it is surprising that we could find only a few publications describing a location-aware DAC-variant.

Wullems (2004) proposed a location-aware variant of the well-known *Access Control List (ACL)* model already explained above. In this model the ACL assigned to an object consists of several "ACL entries". Such an entry is the collection of all the permissions a particular subject has on the respective object. These permissions can have a location constraint described by a polygon. If the user is outside this polygon he cannot perform the operation on the object described by the respective permission.

Leonhardt & Magee (1998) proposed another location-aware DAC variant. However, their model is tailored to tackle the problem of *location privacy*, i.e., to describe who is allowed to query a located user's location data. A survey on the problem of location privacy can be found in Decker (2008e); in this article several scenarios how an attacker can exploit the knowledge of the location of a mobile user are described; further, this survey article also presents several technical methods to thwart such attacks.

While in conventional access control models there is one rule for each access rule, Leonhardt & Magee introduce a second object as a rule's target which is a location. The subject of such a rule is the user who wants to query another user's location information; the first object is that user, whose location information might be accessed. The second object is the location at which the subject has to stay to be allowed to get access. An example from the original paper for such a rule is the following:

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

This rule says that subject "Joe" is allowed to check if user "Fred" is in his vicinity, but only while Joe is in one of the buildings of the school.

In Decker (2008b) an ACM is described that follows the metaphor of digital documents. Each document belongs to exactly one document class, e.g., document class "customer note" could have the instances "customer note #1", "customer note #2" etc. The operations a given user can perform on a given document instance (e.g., read, write, append, delete) can be restricted to particular locations. These permissions can be altered at runtime for each document instance (by a user having the permission to do this). The initial permissions for a document are obtained from its document class at the time of creation. Different document classes only differ in the default permissions they assign to their instances.

Based on these default permissions various application scenarios based on digital documents can be realized, e.g., for a document acting as location-aware *personal reminder note* only the user who created it has permissions on the document. For a document that acts as "virtual graffiti" the creator as read and write permission, but all other users have only "write permission". However, it is at the discretion of the creator to grant write permission to other users. Also a location-aware Wiki is thinkable: a page in such a Wiki gives information pertaining to the location where that page is accessible, e.g., a description of a monument or building at that location. Following the well-known principle of Wikis every user has read as well write access to every page.

A further location-aware DAC model is the one by Gallagher (2002), which is for database management systems (DBMS). This model will be explained in subsection 3.5 which is devoted to ACMs for DBMS.

3.3 Location-aware Mandatory Access Control

Meanwhile there are implementations of MAC outside the domain of the military and secret services available, but this approach of access control still doesn't enjoy a widespread adoption in the civil domain. So it is no wonder that we could only identify a few location-aware MAC models, which will be presented in this subsection:

Ray and Kumar (2006) propose a location-aware variant of MAC. In their model security levels are not only assigned to users and resources, but also to locations. For example, an ordinary office room might be classified as "Confidential", whereas a strong room equipped with an alarm system is classified as "Top Secret". It is demanded that a location that lies within another location has at least the security level of the outer location, e.g., one room in a building classified as "Secret" might be classified as "Top Secret" but not the other way round. A further rule in this model is that resources can be only stored at a location when the security level of the location is not lower than that of the resource, e.g., a document with level "Top Secret" cannot be located in a building with a classification of just "Confidential".

Another location-aware MAC model is the one proposed in Decker (2009b); however, this is a non-generic model, which can only be used for database management systems (DBMS). This model is therefore presented in subsection 3.5, which is devoted to ACMs for DBMS.

3.4 Location-aware Access Control for Mobile Workflow Management Systems

A *Business Process* (or just "Process") is the set of activities that has to be performed to reach a particular goal. Such a goal could be the fulfillment of an order received by a customer. In most cases the activities have to be performed in a particular order; some activities might be optional. It is also possible that sets of activities can be performed parallel. Usually these activities have to be performed by different actors. A *Workflow Management System (WfMS)* is a special information system that supports the definition, execution, simulation and monitoring of business processes (Oberweis, 2005). The part of a business process that is executed by a WfMS is called "workflow". Modern WfMS support the definition of workflows by graphical tools. The execution of workflows includes that the *Workflow Engine* of the WfMS assigns activities to individual user/actors of the WfMS and provides the data necessary for the execution of these activities to the respective actors. Actors of a WfMS usually have a *Workflow Client* that displays in form of a list the activities that are assigned to them; this workflow client can also be used to retrieve the data to perform an activity.

We talk about a *Mobile Workflow* if there are activities in the workflow instances that have to be performed with mobile computers (Decker et al., 2009). Typical examples for such activities are activities that have to be performed at customer's premises (e.g., enter order by customer, inquiry latest price information, consult technical documentation or service history for a machine to repair) or on business journeys. In the academic literature some descriptions of WfMS especially tailored for mobile workflows can be found (e.g., Jing et al., 2000, or Alonso et al, 1996); however, these systems do not have special ACMs.

Some authors proposed ACMs especially for workflow systems (e.g., Wainer et al., 2003; Bertino et al., 2001). One particular feature of these models is that they express requirements concerning the different actors that perform the different steps of a workflow, e.g., that the actor who performed the step "make proposal" in an approval workflow is not the same actor who performs the "make decision" step (Separation of Duties, SoD; Sandhu, 1990). The opposite principle is called "Binding of Duties" (BoD) which means that the actor who performed a particular activity of a workflow instance has also to perform one or more other activities of the same workflow instance (Wainer et al., 2003). The standard example to motivate BoD is that the employee who received a customer's order via the telephone should also make all following contacts during the processing of that order to the customer, so the principle "one face to the customer" is obeyed.

In Decker (2008a) an ACM model is sketched, that is location aware and also process aware. In this model different activities of a workflow can be restricted to particular locations. Considering the example of a workflow dealing with "facility management" the activities "dispatch service technician" and "write bill" could be restricted to the back office while the activity "write onsite report" has to be performed when the respective actor resides at the place where something has to be repaired. It is further distinguished if a location-constraint is assigned at the schema or the instance level: when assigned at the schema level then this restriction holds for all workflow instances. Another idea are dynamic constraints where location-constraints are not defined in advance (at administration time) but rather during runtime: the location where a particular activity is performed is evaluated to restrict the location of another activity. If the activity "repair" for a workflow instance was performed in a particular street then particular activities like "on-site report" or "follow-up inspection" also have to be performed at that location (binding of locations). The opposite case would be a rule that forbids to perform two activities of the same workflow instance at the same location (separation of locations), e.g., for an approval process it could be reasonable to demand that the activities "enter approval" and "make decision" take place at different locations to prevent collusions between employees to obscure fraud or carelessness. "Separation of Locations" and "Binding of Locations" are the transfer of the well-known security principles of "Separation of Duties" and "Binding of Duties" to consider the spatial dimension.

To support the management of mobile workflows there is also a proposal for an extension of *Activity Diagrams* which are part of the OMG's *Unified Modeling Language* (UML; OMG, 2007), which can be found in Decker (2009c; 2009d). An example for an activity diagram can be found in Figure 2 in the upper part which is denoted as "workflow graph". The basic idea to introduce location-aware access control into workflow diagrams is to assign different kinds of location constraints to the activities. These constraints either define where the activity *has to* *be performed* or where the activity *is* <u>*not*</u> *allowed to be performed*, so there are *positive* and *negative* location constraints. Further, there are *direct* and *indirect* location constraints, which will be explained in the following two paragraphs:

Direct constraints are statements about concrete locations where an activity has to be performed (positive constraint) or is not allowed to be performed (negative constraint); they therefore hold for all workflow instances of a workflow definition (schema) and are defined at design time of a the workflow schema before the first instance for that schema is created. Positive constraints could be motivated by considerations that particular activities require special equipment so it is only plausible to perform them at locations where this equipment is available. It is also thinkable that some activities should only be performed at locations that are deemed as "secure" (e.g., company building, trustworthy countries, laboratory) because they require the access to sensitive data or should be performed under the supervision of senior employees. Negative constraints can be used when it would require more efforts to enumerate the locations where some activity is allowed then to enumerate where this activity is not allowed. An example for the application of negative constraints would be a company fearing industrial espionage in a few countries and therefore wants to prohibit that activities which require access to sensitive data (e.g., technical documents, price calculations) are performed in those countries.

Indirect constraints just describe how the actual location for the constraint has to be derived during the runtime of a particular workflow instance. One method to derive the location for a constraint during the runtime of an instance is to employ *location rules*: such a rule says that the *target activity* (the activity that has to get the location constraint) either has to be performed at the same location or is not allowed to be performed at the same location as the *triggering activity* of that rule. To specify what is "the same location" a type of location (location class, e.g., country, region, building, sales district, department) or radius has to be specified. Another approach is to have an external information system that can deliver the location for the constraint upon request. An example for such an information system would be a *Customer Relationship Management (CRM)* system that stores the addresses of all customers of a company. If the purpose of a workflow instance is to fulfill the order of a customer and this workflow includes activities which require visiting that customer's home/premises then the location constraint for the respective activities could be queried from the CRM. Finally, an indirect location constraint could also be specified manually by a human operator during the runtime of a workflow. For example, if a call center operator receives a customer's call that requires sending service to the customer's home then the human operator could define location constraints for the on-site activities of that workflow based on the knowledge of the customer's residence.

To exemplify this description an activity diagram with such location constraints can be found in Figure 2. The upper part of the figure denoted as "workflow graph" (without the dotted lines) represents a conventional activity diagram. After the initial activity *A* either activity *B* or *C* is executed, but never both. Activities *D* and *E* are executed parallel. After the completion of these two parallel activities the final activity *F* is executed before the end of the workflow instance is reached. A possible instance of this workflow schema, in which the optional activity *C* is chosen, is indicated by the bold arrows.

In the lower part of the diagram there are three location constraints which are assigned with dotted arrows to activities of the workflow graph. The constraint assigned to activity *E* is the only direct constraint depicted: it is a negative constraint pointing to "Country X". The mode of the constraint, i.e., if it is a positive or a negative one, is indicated by the symbol in the circle on the dotted arrow that is assigned to the activity. So the meaning of this constraint is that it forbids that for any workflow instance activity *E* is performed by an actor who stays in country X. Activity *A* is the trigger activity for a location rule that assigns a positive location constraint to activity C. The granularity of the rule is the city, i.e., activity *C* has to be performed in the same city where activity A was performed. If the current location of the actor during the execution of the target activity *A* isn't within any city then no location constraint is generated. Finally, there is also an indirect constraints assigned to activity *F*. This constraint is a negative one that obtains the location from an external application.

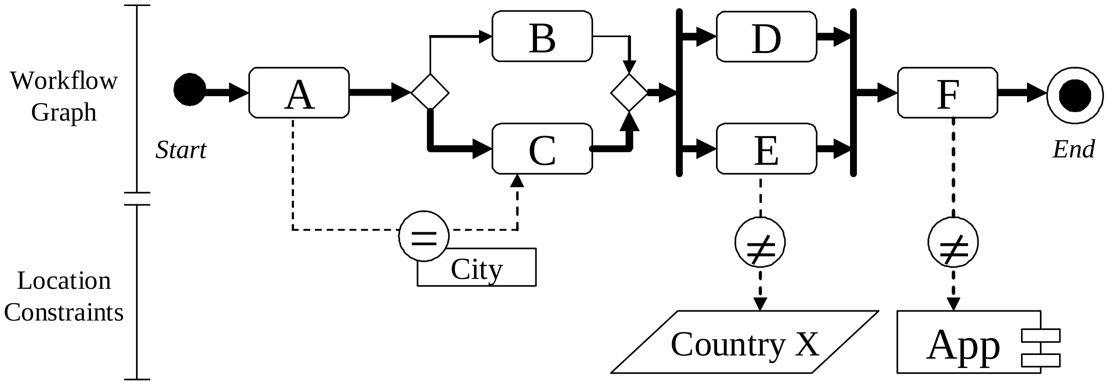

Figure 2. UML Activity Diagram with Location Constraints

It is not only possible to assign location constraints to single activities but also to so called "swimlanes". Swimlanes in UML activity diagrams are used to define subsets of activities to assign them to an individual actor or organizations. Further, the modeling approach also supports to assign one given location constraint to several activities (so called "shared location constraint").

Hewett & Kijsanayothin (2009) also worked on the field of location-aware ACMs for workflows. According to their approach it is possible to assign location constraints to individual activities of a workflow description which say that particular activities can only be performed at that location. Such a location constraint in their model is depicted by a little rectangular box that is attached to the box with rounded edges which represents an activity in UML; this rectangular box contains a textual description of the location or an "*" (asterisk) if the respective activity can be performed every. However, they do not have an elaborated location model for this. Further, it is possible to define rules to forbid the execution of two or more activities of the same workflow instance by the same actor; this way the security principle "Separation of Duties" can be realized. Since actors can also have location constraints it is possible that for a workflow instance for a particular activity no eligible actor is available, because all actors which would be allowed to perform that activity according to their roles are not allowed to perform that activity according to the separation of duties and location constraints. Hewett & Kijsanayothin therefore concentrate on the elicitation of an algorithm to detect such constellations in advance.

3.5 Access Control Models for Database Management Systems

The *Structured Query Language (SQL)* is a language to work with *Database Management Systems (DBMS)*, e.g., to enter data into tables, to update individual table rows or to retrieve data (Elmasri & Navathe, 2004). It supports also administrative tasks like the creation of user accounts and assignment of permissions to individual users.

Gallagher (2002) describes an extension to SQL to enable the location-aware assignment of permissions to users. Using this extension the administrator can grant the right to perform a particular operation on a given table, but only when the user stays within a particular area. An example for such a statement would be as follows:

```
          GRANT select, update ON customers TO alice INSIDE
area1
```

This statement gives user Alice the permission to perform the operations “select” and “update” on the table “customers” when she stays at “area1”. The novel construct is the inside-clause at the end of the statement. A further novel construct of Gallagher’s SQL is the DENY-command that can be used to explicitly deny a permission at a certain region, e.g.

```
DENY select ON customers FROM alice INSIDE area2
```

However, the author does not comment how the geometric locations behind the identifiers for areas can be resolved or which location-model they use.

Another LAACM for DBMS follows the concept of MAC and can be found in Decker (2009b). The basic idea is that individual table rows can “remember” the location where they were created with SQL’s “insert”-statement. Subsequent accesses on these rows are denied if the mobile user stays outside the location where this row was created. For “select”-statements the denial of an access means that the respective row is just hidden (i.e., it is simply excluded from the result set), while for update and delete statements an error message is raised and the execution of the respective command in aborted.

This ACM can be configured for individual tables with individual granularities: one table could remember the countries where each of its rows was created, while another table could remember the city of creation. It is also possible to have tables in a database instance which aren’t location aware at all.

A further feature of this model is that security level can be assigned to locations, e.g., a secured building of the company could have the security level “Top Secret”, which allows to access highly confidential data while staying within that building; in contrast to this a country with a high level of industrial espionage could have the level of just “public”, which prohibits to access any data classified higher than “public” while staying in that country. Further, there are different unordered categories of security levels which correspondent to things like product categories, technologies, or projects. Such a thematic category could be “nuclear technology” or “semi-conductor technology”. A given country might have the security level of “Top Secret” for the first category, while it is only classified as “Confidential” for the latter category. The model’s feature called “indirect location constraints” means that a database table can be configured to “remember” the security level according to a particular category where a data row was inserted into the table. Subsequent accesses on that row will only be permitted if the user stays at a location that is classified at least as high as the row's security level for that category.

An area of application for this ACM would be the enforcement of the policy that personal data of customers shouldn't leave the country where that data was acquired; or that data about a customer should only be accessible in that sales district were that customer has his residence and where the data was acquired.

4. TAMPER-PROOF LOCATING-TECHNOLOGIES

There are many technologies available for determining the location of a mobile computer, see Roth (2004) or Küpper (2007) for an overview. When an access control decision is based on the mobile user's current position this raises the question if it is possible to manipulate the locating process, because if the employed locating system can be manipulated then it is also possible to circumvent the enforcement of location-aware access restrictions. If the legitimate/illegitimate possessor of the mobile device or another attacker is able to affect the locating process this is called "location spoofing". Spoofing is a more serious problem than just performing a *"Denial-of-Service"-attack (DoS-attack)* by jamming the respective signals because the victim might not be aware of the attack, so he might get piloted into an ambush. A reference monitor for a location-aware ACM could deny just every access attempt if the location system is currently out of order because of an ongoing DoS-attack.

There are several articles dealing with special measurements to prevent spoofing. Due to space limitations we can only sketch the basic principles of such measurements. For more details we refer the reader to the survey paper by Decker (2009a) on this topic. It should be also mentioned that anti-spoofing technologies are not required if LAAC is employed as way to support usability of a mobile application by hiding unnecessary information and options from the display.

Location Keys: This technique is based on some kind of information that is only available at a specific location, e.g., locally emitted radio signals carrying random bit sequences or the unpredictable distortion pattern of globally broadcasted signals. A mobile device has to provide this information to a backend system that compares this information to the information it receives form a trusted reference station in the proximity of the alleged location. To prevent a so called "wormhole attack" where a colluding user forwards the information to the attacker, the mobile user is obliged to forward the location key within a certain time span because the forwarding causes additional time delay. An example for an anti-spoofing system based on location keys is called *CyberLocator* and discussed in Denning & MacDoran (1996). In this system as location keys signals are evaluated that are not transmitted for the purpose to prevent location spoofing.

Tamperproof Hardware: Some authors propose locating systems that are secured against location spoofing because the components on the mobile device that are responsible for the calculation of the location are embedded in tamper-proof hardware modules. Such tamper-proof hardware is also a prerequisite if the information that is protected by LAAC is stored on the mobile device rather then on a stationary backend server. An example for this approach to prevent spoofing is the work of Mundt (2006).

Request-Response-Protocols: This family of anti-spoofing techniques is based on the fact that the wireless signals used by a locating technology travel with a certain velocity (e.g., speed of light for radio waves). So if a mobile device claims to be at a particular location it must be able to answer signals emitted by a trusted base station in the proximity to that location within a certain period of time. In an article by Sastry et al. (2003) this principle is used to verify the alleged location of mobile computers. If radio waves are used even little measurement errors induce a great spatial uncertainty, so Sastry et al. use ultrasonic waves for one way of the protocol because these waves travel at a relatively low speed.

Sanity Check: Systems based on this approach perform a plausibility check on the locating signals received (low-level) or the hereof calculated location (high level). For low-level signals it is suspicious if sudden increases of the signal strength are detected because this typically occurs if an attacker sends strong signals to overlay the original signals. A high-level plausibility check would be to simply check if the mobile devices move at a reasonable speed or if it travels through places that are not passable (e.g., building). Several ideas to perform such sanity checks to secure GPS are elicited by Warner & Johnston (2003). It is also an idea to employ several locating systems at the same time (e.g., GPS together with Cell-ID locating), because it is harder to manipulate several locating systems at the same time in a consistent way.

Radio technology-based: There are special low-level techniques to generate radio waves in a way that hardens them against jamming (“deletion of signals”) or manipulation. These techniques include spread-spectrum techniques or coding techniques like the so called *Manchester Coding* (Capkun et al., 2007). Spread spectrum techniques also harden the signal against interferences/jamming and can also be employed as multiplexing technique. It is also harder to manipulate a locating system that sends its navigation messages on several frequencies, e.g., in the Russian “Glonass”-System each satellite has its own frequency whereas all GPS satellites share the same frequencies.

5. EXISTING APPLICATIONS OF LAAC

Some simple forms of LAAC are already found in the real world. They can be seen as precursors of more advanced forms of LAAC envisioned by the authors of the LAACMs presented in section 3.

Personal Navigation Devices (PND) are mobile computers that help travelers (e.g., motorists, hikers) to find the way to a particular destination. These devices are connected to a GPS receiver. There are PND available that can be activated only after entering a secret number (PIN) to deter thievery. This PIN can only be reset when the device is at the location where the PIN was set (e.g., Garmin Nüvi series).

The standard for *Digital Versatile Discs (DVD)* comprises the so called “region code”. According to this code the world is divided into eight different regions (e.g., USA belongs to “region 1”, western and central Europe are parts of “region 2”). DVD players should have a built-in region-code according to the country where they are sold and play only discs whose region code matches the built-in code. This system is motivated by the fact that different countries have different laws for the protection of the youth and a distributor for a movie may buy the right to sell a movie only in certain countries.

Internet Protocol (IP) addresses can be roughly allocated to a certain geographic region. This is used by some websites to personalize the contents according to the origin country of the respective user, e.g., by adapting the language or by showing advertisement offers concerning the region of the user. IP locating is also used by some websites to restrict access to their content to users from certain countries. One example is *hulu.com*, a portal that provides streaming of selected movies and current TV shows but restricts the access to requests originating from the USA. However, by using proxy servers this restriction can be easily circumvented.

6. FUTURE RESEARCH DIRECTIONS

Despite all the works on the field of LAAC we discussed so far in this article we identified areas where spending effort on further research work would be worthwhile (see also Decker, 2008d):

The majority of LA ACMs are extensions for RBAC. However, considering generic models it would be interesting to have more LA DAC models as well as LA MAC models; especially the former is of interest since DAC is the prevalent ACM used in contemporary software systems. Further, we think that many novel application-specific ACMs with location-awareness could increase the security of the mobile employment of these systems.

For the management of LA ACMs it is necessary to have appropriate software tools. Such a tool should support working with geographic maps for the definition and visualization of spatial extends. A further feature of such tools should be the detection of inconsistencies in location-aware access control rules. However, so far only rudimentary tool support for LAAC models can be found in the research literature (e.g., Decker, 2008c; Bhatti et al, 2008; Cruz et al., 2008).

For real world applications it would also be necessary that a LAACM can state requirements concerning the employed locating system, e.g., that particular permissions should only be granted when the locating system is tamper-proof or can guarantee to determine the user's location with a certain degree of accuracy or reliability.

Negative permissions – i.e. the definition of a location where something is explicitly forbidden rather the defining where something is explicitly allowed – are a feature that is only offered by two models we presented in our survey. However, if a model supports positive as well as negative permissions this could lead to inconsistencies, e.g., if an activity has to be performed in Berlin (positive constraint) but at the same time is not allowed to be performed within Germany (negative constraint).

7. CONCLUSION: SUMMARY AND OUTLOOK

In our article we first motivated the usefulness of location-aware access control by several examples. Afterwards we introduced some basics concerning access control before we gave an overview about several access control models which allow formulating statements concerning a user's current position a condition to allow an access attempt. We also sketched different principles to avoid the manipulation of locating systems, because tamper-resistant locating systems are the base for the enforcement of location-aware access control policies.

Our literature survey shows that the majority of LAACM models are based on RBAC, so further research on location-aware MAC and DAC would be interesting. Also, we think there is a lot of potential for application-specific ACMs with location-awareness. Further, it would be worthwhile to develop special tools for the management of location-aware ACMs.

REFERENCES

Alonso, G., et al., 1996. Exotica/FMDC: A Workflow Management System for Mobile and Disconnected Clients. *In Distributed and Parallel Databases*. Vol. 4, No. 3, pp. 229-247.

Bell, D.E., 2005. Looking Back at the Bell-La Padula Model. *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC 2005)*, Tucson, USA, pp. 337-351.

Bertino, E., Bonatti, P.A., and Ferrari, E., 2000. TRBAC: A Temporal Role-based Access Control Model. *Proceedings of the 5th ACM Workshop on Role-Based Access Control (RBAC 2000)*, Berlin, Germany, pp. 21-30.

Bertino, E., Ferrari, E., and Atluri, V., 1999. The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. *In ACM Transactions on Information and System Security*. Vol. 2, No. 1, pp. 65-104.

Bhatti, R., Damiani, M.L., Bettis, D.W., and Bertino, E., 2008. Policy Mapper: Administering Location-Based Access-Control Policies. In *IEEE Internet Computing*, Vol. 12, No. 2, pp. 38-45.

Benantar, M., 2006. *Access Control Systems. Security, Identity Management and Trust Models*. Springer, Heidelberg, Germany et al., 2006.

Capkun, S., Bonne, K., and Cagalj, M., 2007. SecNav: Secure Broadcast Localization and Time Synchronization in Wireless Networks. *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '07)*, Montréal, Canada, pp. 310-313.

Chandran, S.M., and Joshi, J.B.D., 2005. LoT-RBAC: A Location and Time-Based RBAC Model. *Proceedings of Conference on Web Information Systems Engineering (WISE 2005)*, , New York, USA, Springer, pp. 361-375.

Cruz, I.F., et al., 2008. A Constraint and Attribute Based Security Framework for Dynamic Role Assignment in Collaborative Environments. *Proceedings of CollaborateCom*, Orlando, USA, pp. 322-339.

Damiani, M.L, Bertino, E., and Perlasca, P., 2007. Data Security in Location-Aware Applications: An Approach Based on RBAC. *International Journal of Information and Computer Security*. Vol. 1, No. 1/2, pp. 5-38.

Decker, M., 2008a: A Security Model for Mobile Processes. *International Conference on Mobile Business (ICMB ‘08)*, IEEE, Barcelona, Spain.

Decker, M., 2008b: Location-Aware Access Control for Mobile Information Systems. *Collaboration and the Knowledge Economy (Proceedings of eChallenges 2008)*. Stockholm, Sweden, IOS Press, pp. 1273-1280.

Decker, M., 2008c. An Access-Control Model for Mobile Computing with Spatial Constraints. *Proceedings of the International Conference on e-Business (ICE-B 2008)*, INSTICC, Porto, Portugal, pp. 185-190.

Decker, M., 2008d. Requirements for a Location-Based Access Control Model. *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia (MoMM)*, Linz, Austria, ACM, pp. 346-349.

Decker, M., 2008e: Location Privacy - An Overview. *International Conference on Mobile Business (ICMB ‘08)*, IEEE, Barcelona, Spain.

Decker, M., 2009a. Prevention of Location-Spoofing. A Survey on Different Methods to Prevent the Manipulation of Locating-Technologies. *Proceedings of the 7th International Joint Conference on e-*

Business and Telecommunications. International Conference on e-Business (ICE-B), Milan, Italy, pp. 109-114.

Decker, M., 2009b. Mandatory and Location-Aware Access Control for Relational Databases. *Proceedings of the International Conference on Communication Infrastructure, Systems and Applications in Europe (EuropeComm 2009)*. London, U.K., Springer LNICST, pp. 217-228.

Decker, M., 2009c. An UML Profile for the Modelling of mobile Business Processes and Workflows (Article No. 38). *Proceedings of the 5th International ICST Mobile Multimedia Communications Conference (MobiMedia)*, Kingston upon Thames, U.K., ACM.

Decker, M., 2009d. Modelling Location-Aware Access Control Constraints for Mobile Workflows with UML Activity Diagrams. *Proceedings of the Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UbiComm 2009)*, Sliema, Malta, IEEE, pp. 263-268.

Decker, M., 2009e. A Location-Aware Access Control Model for Mobile Workflow Systems. In *International Journal of Information Technology and Web Engineering (IJITWE)*, Vol. 4, No. 1, pp. 50-66.

Decker, M., Stürzel, P., Klink, S., and Oberweis, A. (2009). Location Constraints for Mobile Workflows. Techniques and Applications for Mobile Commerce. *Proceedings of the Conference on Techniques and Applications for Mobile Commerce (TaMoCo '09)*. Mérida, Spain, pp. 93-102.

Denning, D.E., and MacDoran, P.F., 1996. Location-Based Authentication: Grounding Cyberspace for Better Security. *In Computer Fraud & Security*, No. 2, pp. 12-16.

Elmasri, R., & Navathe, S., 2004. *Fundamentals of Database Systems (4th Edition)*. Pearson, Boston, USA.

Ferraiolo, D.F., Kuhn, D.R., and Chandramouli, R., 2007. *Role-Based Access Control (Second Edition)*. Artech House, Norwood, USA.

Gallagher, M., 2002: *Location-Based Authorization*. Master Thesis supervised by Shashi Shekhar, University of Minnesota, USA.

Hansen, F., and Oleshchuk, V., 2003. SRBAC: A Spatial Role-Based Access Control Model for Mobile Systems. *Proceedings of NORDSEC 2003*, Gjovik, Norway, pp. 129-141.

Hewett, R., and Kijsanayothin, P., 2009. Location Contexts in Role-based Security Policy Enforcement. *Proceedings of the 2009 International Conference on Security and Management (SAM'09)*, Las Vegas, USA, pp. 404-410.

Jing, J., Huff, K., Hurwitz, B., Sinha, H., Robinson, B., and Feblowitz, M., 2000. WHAM: Supporting Mobile Workforce and Applications in Workflow Environments. *Proceedings of 10th International Workshop on Research Issues in Data Engineering (RIDE)*, San Diego, USA, pp. 31-38.

Küpper, A., 2007. *Location-based Services — Fundamentals and Operation (Reprint)*. Wiley & Sons, Chichester, U.K.

Liao, H.-C., and Chao, Y.-H., 2008. A New Data Encryption Algorithm Based on the Location of Mobile Users. *In Information Technology Journal*, Vol. 7, No. 1, 2008, pp. 63-69.

Lampson, B., 1974. Protection. *In Operating Systems Review*, Vol. 8, No. 1, pp. 18-24.

Leonhardt, U., and Magee, J. (1998). Security Considerations for a Distributed Location Service. *In Journal of Network and Systems Management*, Vol. 6, No. 1, pp. 51-70.

Moyer, J.M, and Ahamad, M., 2001. Generalized Role-Based Access Control. *Proceedings of the 21st International Conference of Distributed Computing Systems.* Mesa, USA, IEEE, pp. 391-398.

Mundt, T., 2006. Two Methods of Authenticated Positioning. *Proceedings of the International Workshop on Quality of Service & Security for Wireless and Mobile Networks (Q2SWinet)*, Terromolinos, Spain, pp. 25-32.

Oberweis, A., 2005. Person-to-Application Processes. Workflow-Management. Chapter 2 in *Process-Aware Information Systems — Bridging People and Software through Process Technology*. Wiley, Hoboken, USA, pp. 21-36.

OMG, 2007. *Unified Modeling Language (OMG UML), Superstructure, V2.1.2*. Object Management Group (OMG), Needham, USA, 2007.

Ray, L., Kumar, M., 2006. Towards a Location-based Mandatory Access Control Model. *In Computer & Security*, Vol. 25, No. 1, pp. 46-44.

Ray, I., Kumar, M., and Yu, L., 2006. LRBAC: A Location-Aware Role-Based Access Control Model. *Proceedings of ICISS (LNCS 4332)*, Kolkata, India, pp. 147-161.

Ray, I., and Toahchoodee, M., 2007. A Spatio-temporal Role-Based Access Control Model. *Data Applications and Security* (LNCS 4602), Redondo Beach, USA, pp. 211-226.

Roth, J., 2004. Chapter 7: Data Collection. *Location-based Services*. Morgan Kaufmann, Amsterdam, Netherlands, pp. 175-205.

Sandhu, R.S., 1990. Separation of Duties in Computerized Information Systems. *Results of the IFIP WG 11.3 Workshop on Database Security (DBSec)*, Halifax, U.K., pp. 179-190.

Sastry, N., Shankar, U., and Wagner, D., 2003. Secure Verification of Location Claims. *Proceedings of the 2nd ACM Workshop on Wireless Security (WiSE '03)*, San Diego, USA, pp. 1-10.

Wainer, J., Barthelmess, P., and Kumar, A., 2003. W-RBAC: A Workflow Security Model Incorporating Controlled Overriding of Constraints. *In International Journal of Cooperative Information Systems*, Vol. 12, No. 4, pp. 455-485.

Warner, J.S., and Johnston, R.G. (2003). GPS Spoofing Countermeasures. *Technical Report No. LAUR-03-6163*, Los Alamos National Laboratory, Los Alamos, USA.

Wullems, C.J. (2004). *Engineering Trusted Location Services and Context-Aware Augmentations for Network Authorization Models*. PhD Thesis, Faculty of Information Technology, Queensland University of Technology, Australia.