1 Managing Privacy in Location-based Access Control Systems

Claudio Agostino Ardagna, Marco Cremonini, Sabrina De Capitani di Vimercati and Pierangela Samarati

Dipartimento di Tecnologie dell'Informazione - Università di Milano, Italy

1.1 INTRODUCTION

Preserving user data privacy is one of the hottest topics in computer security. Security incidents, faulty data management practices and unauthorized trading of users personal information have often been reported in recent years, exposing victims to ID theft and unauthorized profiling [46]. These issues are raising the bar of privacy standards, fostering innovative research, and driving new legislations. Some approaches aimed at privacy protection deal with minimizing unnecessary release of personal information or focus on preventing leakage of personal information while in transit or once it has been released to an authorized party, for example, by delayed enactment of privacy preferences [49]. Our work addresses the latter concern in the framework of location-based services. We consider privacy requirements for Location-Based Access Control (LBAC) systems that require, for the provision of an online service, to evaluate conditions depending on users physical locations [6]. In the LBAC area, privacy has been mostly addressed by developing models and techniques that let users access anonymously to online services [10, 12, 23]. Solutions providing different degrees of privacy according to user preferences or business needs are instead less explored. For instance, obfuscation techniques applied to user locations are well-suited to degrade the location accuracy for privacy reasons. In this context, however, only solutions based on increasing the granularity of a location measurement have been investigated and implemented in practice [23, 44]. Moreover, the importance of striking a balance between obfuscating locations for privacy reasons and preserving an acceptable accuracy for LBAC policies evaluation is often mentioned but not yet fully supported. In particular, key for managing such contrasting requirements is the availability of a metric (*relevance*, in our work) measuring, at the same time, the achieved privacy level and the required location accuracy. This metric should be

independent from technological details of location measurements and from LBAC systems peculiarities. This way privacy and accuracy requirements can be evaluated, negotiated, compared, and integrated in a coherent framework.

In this chapter, after a discussion on related work (Section 1.2), we present the scenario and concepts (Section 1.3) that are at the basis of our location-based access control system and authorization language (Section 1.4). We also describe obfuscation techniques that modify location information to provide user privacy protection (Section 1.5). Finally, we illustrate a privacy-aware location-based access control system that integrates the obfuscation techniques (Section 1.6) and we conclude the chapter (Section 1.7).

1.2 RELATED WORK

Works related to location privacy techniques can be categorized into three main classes: *anonymity-based*, *obfuscation-based*, and *policy-based*.

Anonymity-based techniques provide solutions for the protection of the identities of the users. This class includes all solutions based on the notion of *anonymity* [10, 12, 19, 23], which is aimed at making an individual (i.e., her identity or personal information) not identifiable. Beresford and Stajano [10, 11] propose a method, called *Mix zones*, based on an anonymity service that delays and reorders messages from subscribers within pre-defined zones. The proposal is based on a trusted middleware that lies between the positioning systems and the third party applications, and is responsible for limiting the information collected by applications. The Mix zones model introduces the concepts of *application zone*, which are homogeneous application interested located in a specific geographic area, and *mix zones*, which represent areas where a user cannot be tracked. In particular, within mix zones, a user is anonymous in the sense that the identities of all users coexisting in the same zone are mixed and become indiscernible. Furthermore, the infrastructure makes a user entering the mix zone unlinkable from other users leaving it. The Mix zones model is aimed at protecting long-term user movements still allowing the interaction with many location-based services. Other works [19, 23] are based on the concept of location k-anonymity, meaning that a user is indistinguishable from other $k-1$ users in a given location area or temporal interval. Gruteser and Grunwald [23] define k-anonymity in the context of location obfuscation. They propose a middleware architecture and an adaptive algorithm to adjust location information resolution, in spatial or temporal dimensions, to comply with the specified anonymity requirements. Finally, another strand of research is aimed at protecting the *path privacy* of the users [22, 24, 29]. Path privacy involves the protection of users that are continuously monitored during a time interval. This research area is particular relevant for location tracking applications, where data about users moving in a particular area are collected by external services, such as navigation systems, that use them to provide their services effectively. In summary, anonymity-based techniques are suitable for all those contexts that do not need knowledge of the identities of the users, and their effectiveness strongly depend on the number of users physically located in the same area.

Obfuscation-based techniques provide solutions for the protection of location privacy. This class includes all the solutions based on the notion of *obfuscation* [3, 7, 9, 15, 44], which is the process of degrading the accuracy of the location information to provide privacy protection. Differently from anonymity-based techniques, the main goal of obfuscation-based techniques is to perturb the location information still maintaining a binding with the identities of users. Duckham and Kulik [15, 16] define a framework that provides a mechanism for balancing individual needs for high-quality information services and for location privacy. The proposed solution is based on the *imprecision concept*, which indicates the lack of specificity of location information. The authors propose to degrade the quality of location information and to provide obfuscation features by adding n points, with same probability, to the real user position. Obfuscation-based solutions also provide mechanisms for specifying privacy preferences in a common and intuitive manner (i.e., as a *minimum distance*), which, however, presents several common drawbacks. First, they do not provide a metric for the privacy level, making them difficult to integrate into a full fledged location-based application scenario [6]. Second, they usually implement a single obfuscation technique based on the enlargement of a location area. This way, a possibility that is often neglected by traditional location obfuscation solutions is the definition and composition of different obfuscation techniques to increase their robustness with respect to possible de-obfuscation attempts performed by adversaries. Finally, obfuscation solutions are often meaningful in a specific application context only. Ardagna et al. [3, 5, 7] address the above shortcomings by presenting a novel solution composed by a management process and several techniques aimed at preserving location privacy by artificially perturbing location information measured by sensing technologies. Key aspects of the proposal is, on the one side, to permit the specification of privacy preferences in a simple and intuitive way, and, on the other side, to make the enforcement of privacy preferences manageable for location-based services, while preserving the quality of the online service. To this aim, the authors introduce the *relevance* concept as a metric for the accuracy of location information, abstracting from any physical attribute of sensing technology. This permits to quantitatively evaluate the degree of privacy introduced into a location measurement and is adopted by users to define their privacy preferences. Based on relevance preferences, different obfuscation-based techniques and their composition are discussed.

Policy-based techniques are based on the notion of *privacy policies* [20, 26, 27, 30, 34]. Privacy policies define restrictions that must be followed when the location of users is used by or released to third parties. Key to policy-based techniques is the definition of policies that can rule location management and disclosure. The definition of complex rule-based policies is, however, difficult to understand and manage for users that often are not familiar with specific policy definition languages. Therefore, although policies-based techniques are powerful and flexible, they can easily result in very complex and unmanageable tools for endusers.

Technologies for integrating multiple sources of location information are also investigated [39]. Today, most commercial location platforms include a gateway

that mediates between location providers and location-based applications [44]. In these architectures, the location gateway obtains subscriber's location information from multiple sources and delivers them, possibly modified according to privacy requirements to location-based applications. The increased accuracy, and reliability of location technologies have suggested novel ways to exploit location information within location-based services. Some early mobile networking protocols linked the notion of physical position of a terminal device with its capability of accessing network resources [1]. More recently, an emerging research issue is represented by the inclusion of a negotiation phase of QoS parameters based on SLA agreements and privacy preference in LBS [5, 6]. Widespread adoption of wireless local networks has been the subject of some recent studies focused on location-based information for monitoring users movements, based on Wireless-Lan [17] and 802.11 Networks [18].

Another strand of research focuses on the underlying description of the architecture and operations of an access control server in a LBS context. For instance, the need for a protocol-independent location technique has been explored by Nord et al. [42], which assume heterogeneous positioning sources like GPS, Bluetooth, and WaveLAN for designing location-aware applications. Given such different sources of location information, a generic positioning protocol for interchanging position information between position sources and client applications is introduced and different techniques for merging position information are presented. Another work [52] studies location-based information and its management in the area of mobile commerce applications and presents an integrated location management architecture to support composite location requirements. However, coordination among multiple wireless networks, location negotiation protocols for mobile commerce, and privacy issues are not considered yet.

Few proposals, instead, consider location information as a means for improving security. Sastry et al. [48] exploit location-based access control in sensor networks. Zhang and Parashar [53] propose a location-aware extension to Role-Based Access Control (RBAC) suitable for grid-based distributed applications. Ardagna et al. [6] propose a location-based access control model and language together with an evaluation infrastructure. Other papers take into account time variant information for querying database containing location information [32, 36].

Other works follow a different approach by considering the location information as a resource to be protected against unauthorized access. For instance, in [28], a mechanism to protect user's location information by means of electronic certificates, delegation, and trusted location-based services is described. The same problem is addressed in [26] by proposing a privacy-aware architecture for a global Location Service, which should permit users to define rules for the access to their location information.

Finally, several works propose special-purpose *location middlewares* for managing interactions between applications and location providers, while maximizing the quality of service (QoS) [40, 41, 47]. Typically, in these proposals the location middleware *i)* receives requests from LBS components asking for location information, *ii)* collects users locations from a pool of location providers, and *iii)* produces an answer. Naguib et al. [40] present a middleware framework, called *QoSDREAM*, for

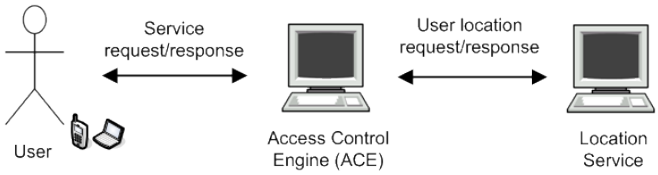

Fig. 1.1 Basic Location-based Access Control Architecture

managing context-aware multimedia applications. Nahrstedt et al. [41] present a QoS middleware for ubiquitous computing environments aimed at maximizing the QoS of distributed applications. Ranganathan et al. [47] present a middleware that provides a clear separation between business applications and location detection technologies. They also address the issue of managing location data from heterogeneous location technologies. Although several middleware components supporting communication and negotiation between location services and applications have been presented, only few proposals try to integrate service quality and privacy protection. For instance, Myles et al. [38] propose an architecture based on a middleware managing the interactions between location-based applications and location providers and on the definition of policies for data release. Hong et al. [30] present an extension of the P3P language for representing user privacy preferences for context-aware applications. Ardagna et al. [3] provide a middleware-based architecture for integrating privacy preferences of the users and location accuracy of LBS in the context of location-based access control systems.

1.3 BASIC SCENARIO AND CONCEPTS

1.3.1 Location-based access control architecture

In a LBAC scenario, there are more parties involved than in conventional access control systems. A LBAC system evaluating a policy does not have direct access to location information; rather, it sends location requests to external services, called *Location Services*, and waits for the corresponding answers [6]. The characteristics of these Location Services will depend on the communication environment where the user transaction takes place. Here, we focus on the mobile network, where Location Service is provided by mobile phone operators. Typically, a LBAC scenario involves the following three entities (see Figure 1.1).

User. The entity whose access request to a service must be authorized by a LBAC system. We make no assumption about users, besides the fact that they carry terminals enabling authentication and some form of location verification.

Access Control Engine (ACE). The entity that implements the LBAC system. It is responsible for evaluating access requests according to some policies containing location-based conditions. The ACE must communicate with a Location

Service for acquiring location information, and it is not restricted to a particular access control model and authorization language. Therefore, an ACE can implement any privacy-aware access control language enriched with location-based conditions.

Location Service (LS). The entity that provides the location information. The types of location requests that it can satisfy depend on the specific mobile technology, the methods applied for measuring users positions, and environmental conditions.

Note that the functional decomposition between the ACE and the LS is due to the fact that location functionalities are fully encapsulated within remote services that are set up and managed by the mobile operators. Therefore, no assumption can be made on these services besides their interfaces.

However, the design of privacy-aware systems poses novel architectural and functional issues that were never considered before in the context of traditional access control systems. Among these issues, the problem of protecting location privacy of users stands out and the need of a privacy-aware LBAC system arises. A privacy-aware LBAC architecture must be designed integrating components logically tied with the applications that need location-based access control enforcement and components providing privacy-aware location services. One typical approach to this problem is to integrate a *Location Middleware* that acts as a trusted gateway between a LBAC system and location services. The Location Middleware should be able to interact with multiple Location Services and to offer location services to an Access Control Engine. It also should manage low-level communications with Location Services and should enforce both privacy preferences expressed by Users and requirements for location accuracy set by an Access Control Engine. The reference privacy-aware LBAC architecture is showed in Figure 1.2.

Communications among logical components are performed via request/response message exchanges. The interaction flow can be logically partitioned in the following six macro-operations.

1. *Initialization*, when user preferences and LBAC policies are defined.
2. *Access request and information negotiation*, when a User submits an access request to the Access Control Engine, and a negotiation process resulting in a bidirectional identification between the parties takes place.
3. *Location service access and SLA negotiation*, when the Access Control Engine requires location information/service to the Location Middleware; a Service Level Agreement (SLA) could be specified to agree upon QoS attributes.
4. *Location information retrieval*, when the Location Middleware collects user location information through a communication process with multiples Location Services.
5. *Location privacy protection*, when obfuscation techniques are used to comply with both user preferences and LBAC accuracy.

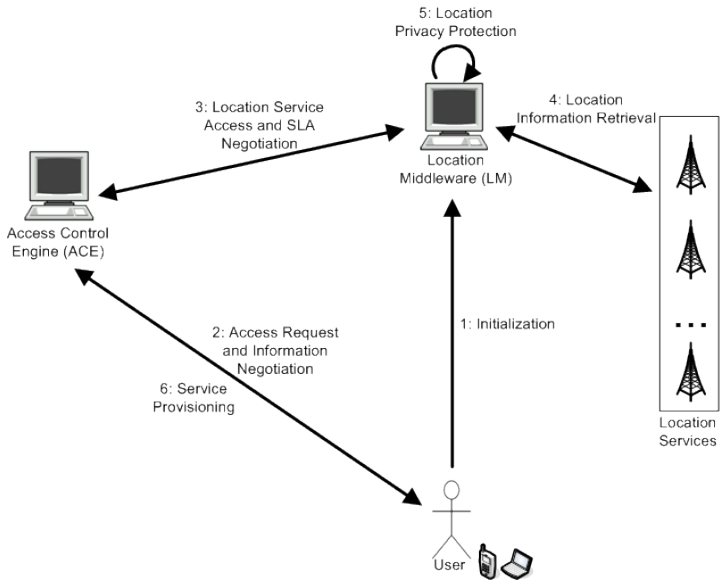

Fig. 1.2 Privacy-aware LBAC Architecture

6. *Service provision*, when LBAC policies are evaluated and the access request is granted or denied.

1.3.2 Location measurements

Two characteristics are specific to technologies for location measurements:

- *interoperability*: location gathering could rely on different sources of location information, depending on availability and cost;
- *accuracy*: each location measurement exhibits a variable accuracy affected by technological limitations (i.e., measurement errors) and possible environmental effects.

While interoperability largely depends on roaming agreements between mobile phone operators and is more business-oriented in nature, accuracy needs to be carefully considered in the design of LBAC systems.

Today, in the mobile network scenario, no technology is available ensuring fully exact user location [31]. The location *accuracy* is always less than 100%, so typically a position is specified as a range, locating the user within a circular area. For a given location request, the location area may depend on the number of nearby antennas and on the surrounding landscape features. Also, a location measurement is often unstable because of changing environmental conditions, such as reflection or interferences that may corrupt the signal. In our model, we take into account these aspects by assuming

that the result provided by a Location Service is always affected by a measurement error. This fact is relevant to the syntax and semantic of the Location Service interface because the outcome of the evaluation of an access request determined by the Access Control Engine will depend on such an uncertainty, which must then be explicitly represented and processed in terms of accuracy.

It is worth noting that suitability and accuracy of a location service largely depend on the underlying technology. GSM/3G technologies are widespread and recent advancements have sensibly improved location capabilities [2]. 802.11 WiFi and AGPS/GPS [21, 45] could also be exploited although some limitations reduce their applicability. WiFi has a limited coverage and its usage is restricted to indoor environments or in urban areas covered by hotspots. GPS, on the contrary, does not work indoor or in narrow spaces but has no coverage limitation, a feature which makes it an ideal location technology for open, outdoor environments.

A direct consequence of such a lack of accuracy is that the location position of a user cannot be expressed as a geographical point. We therefore introduce a first working assumption that considers the shape of a location measurement returned by a Location Service.

Assumption 1 *A location measurement is represented by a planar and circular area.*

This assumption makes the analysis and the design of LBAC systems more tractable with no loss of generality because: *i)* it represents a particular case of the general requirement of considering convex areas (areas must be convex to easily compute integrals over them); *ii)* circular areas approximate well the actual shape resulting from many location technologies (e.g., cellular phones location). In the following we use $Area(r_i, x_i, y_i)$ to state a location measurement centered on coordinates (x_i, y_i) and with radius r_i.

In the same vein of other works in this field [37], we introduce a second assumption as follow.

Assumption 2 *Consider a random location within a location measurement $Area(r, x, y)$, where a "random location" is a neighborhood of a random point ($\hat{x}$,$\hat{y}$)$\in Area(r, x, y)$. The probability that the real user's position (x_u, y_u) belongs to a neighborhood of a random point ($\hat{x}$,$\hat{y}$) is uniformly distributed over the whole location measurement.*

Accordingly, the joint probability density function (pdf) of the real user's position can be defined as follows [43].

Definition 1.3.1 (Joint pdf) *Given a location measurement $Area(r_i, x_i, y_i)$, the joint probability density function (joint pdf) $f_r(x, y)$ is:*

$$f_r(x,y) = \begin{cases} \frac{1}{\pi r^2} & \text{if } x, y \in Area(r_i, x_i, y_i) \\ 0 & \text{otherwise.} \end{cases}$$

1.3.3 Location accuracy

The accuracy of a location measurement returned by a sensing technology depends on the radius of the measured circular area, which, in its turn, depends on the unavoidable measurement error of the location technology. To evaluate the quality of a given location measurement, its accuracy must then be compared to the best accuracy that location technologies are able to provide.

Several works describe and discuss different location technologies and the best accuracy that can be achieved [25, 50]. In [50], the authors provide a survey of standard positioning solutions for the cellular network such as, *E-OTD* for GSM, *OTDOA* for Wideband CDMA (WCDMA), and *Cell-ID*. Specifically, E-OTD location method is based on the existing observed time difference (OTD) feature of GSM systems. The accuracy of the E-OTD estimation, in recent studies, has been found to range from 50m to 125m. Observed Time Difference Of Arrival (OTDOA), instead, is designed to operate over wideband-code division multiple access (WCDMA) networks. The positioning process achieves a location accuracy of 50m at most. Finally, Cell-ID is a simple positioning method based on cell sector information, where cell size varies from 1-3km in urban areas to 3-20km in suburban/rural areas.[1]

Therefore, the accuracy of a measured area corresponds to its radius, which we call r_{meas}. To evaluate the quality of a location measurement, the accuracy of a given measurement must be compared with the best accuracy that the technology can achieve. Therefore, calling r_{opt} the radius representing the best accuracy (i.e., the minimum measurement error), ratio r_{opt}^2/r_{meas}^2 is a good estimation of the quality of a location measurement. As an example of a positioning process using the three technologies described above, suppose that a user position is located with radius r_{meas}=62.5m using E-OTD method, radius r_{meas}=50m using OTDOA, and radius r_{meas}=1km using Cell-ID. The area corresponding to the best accuracy (minimum radius) has r_{opt}=50m. The three location measurements result in three different areas. In particular, the area with best accuracy is provided by OTDOA and has a measurement quality of 1, whereas the others have a quality proportionally reduced to 0.8 for the area calculated by means of E-OTD, and 0.05 for the one measured through Cell-ID. This way, based on the optimal location accuracy, we can distinguish between different measurement accuracies and reward the best technology available.

1.3.4 Relevance

The notion of relevance is strictly related to the notion of accuracy. The relevance is defined as an adimensional, technology-independent metric of the location position accuracy. A location position could be either a location measured by sensing technology or an obfuscated location. The relevance is a value $\mathcal{R} \in (0, 1]$ that:

- tends to 0 when location information must be considered unreliable. This represents the limit condition of very large values of measurement errors or

[1] Other methods (e.g., [13, 14]) are able to further improve the accuracy of standard positioning methods.

obfuscations that degrade the information so that no relation with the original measured location is preserved;

- is equal to 1 when location information has best accuracy. This represents the second limit condition of a measurement error equal to the one introduced by the best sensing technology and no obfuscation applied;

- falls in (0,1) when the location accuracy is less than optimal either for measurement errors larger than the minimum and/or for degradations artificially introduced by obfuscation techniques. This represents the standard situation where the degradation of the original accuracy provides a certain level of privacy as required by users, while keeping, however, an acceptable degree of accuracy as needed by application providers.

Accordingly, the *location privacy* provided by an obfuscated location is evaluated by (1-$\mathcal{R}$). In different terms, the notion of relevance is useful to normalize the accuracy value of a location position by expressing it as an adimensional value, independently from any physical scale or from a reference area if given as a percentage of loss, and to represent the lack of accuracy of obfuscated positions regardless to the specific applied obfuscation techniques. The relevance is the general functional term used to qualify the accuracy (and correspondingly the privacy) of a location position when an LBS interacts with users or application service providers, which, in general, are unaware of the technicalities of both location sensing technologies and obfuscation techniques.

In our reference scenario, an LBS has to manage locations that, on the one side, could be perturbed for privacy reasons, while on the other side could be required to have an accuracy not below a threshold to preserve a certain quality of service. To support such requirements, all location measurements have an associated relevance value and all management decisions, either related to users privacy or to quality of information, are carried out by considering or possibly negotiating relevance values associated with the location measurement.

The following three relevance values characterize our privacy management solution.

- *Initial relevance ($\mathcal{R}_{Init}$).* The relevance of a user location measurement as returned by a sensing technology. This is the initial value of the relevance that only depends on the intrinsic measurement error.

- *Final relevance ($\mathcal{R}_{Final}$).* The relevance of a final obfuscated area produced by satisfying a user's privacy preference. It is derived, starting by the initial relevance, through the application of one or more obfuscation techniques.

- *Required relevance ($\mathcal{R}_{LBAC}$).* The minimum relevance required by an ACE for a reliable evaluation of a location-based policy. This value represents the threshold for the acceptable accuracy of a location measurement or a location predicate evaluation. Below this threshold, the ACE considers the location information too inaccurate for an access control decision.

The value of $\mathcal{R}_{Init}$ is calculated by normalizing the best accuracy that could have been achieved with respect to the technical accuracy resulting from the specific measurement. This is represented by the ratio of two measurement errors: the area that would have been returned if the best accuracy was achieved (i.e., having radius r_{opt}) and the actual measured area (i.e., having radius r_{meas}). In other words, $\mathcal{R}_{Init}$ measures the relative accuracy loss of a given measure - due to, for example, particular environmental conditions - with respect to the best accuracy that the technology would have permitted. This is the only relevance value that is directly calculated from physical values (i.e. measurement errors). $\mathcal{R}_{Final}$ is derived from $\mathcal{R}_{Init}$ by considering the accuracy degradation introduced for privacy reason. We use a scalar factor $\lambda \in (0, 1]$ to represent it. Accordingly, the location measurement associated with $\mathcal{R}_{Init}$ will be perturbed by applying obfuscation techniques so that a resulting area having relevance $\mathcal{R}_{Final}$ is obtained.

Definition 1.3.2 ($\mathcal{R}_{Init}$ and $\mathcal{R}_{Final}$) *Given a location measurement area of radius r_{meas} measured by a sensing technology, a radius r_{opt} representing the best accuracy of sensing technologies, and a degradation $\lambda \in (0, 1]$, initial relevance $\mathcal{R}_{Init}$ and final relevance $\mathcal{R}_{Final}$ are calculated as:*

$$\mathcal{R}_{Init} = \frac{r_{opt}^2}{r_{meas}^2} \tag{1.1}$$

$$\mathcal{R}_{Final} = \lambda \mathcal{R}_{Init} \tag{1.2}$$

Differently, the value of $\mathcal{R}_{LBAC}$ is given, either autonomously defined by the ACE as a requirement for the access control decision, or negotiated as a QoS parameter of the location service.

1.4 LOCATION-BASED ACCESS CONTROL

Conventional access control mechanisms rely on the assumption that requesters' profiles fully determine what they are authorized to do. However, context information and, in particular, physical user locations may also play an important role in determining access rights. We describe the integration of access control policies with location-based conditions, focusing on policy evaluation and enforcement, which represent challenging issues that such an extension to access control policies inevitably carries with. *Location-based Access Control* (LBAC) supports access control policies that include conditions based on the physical location of a requester (e.g., to be inside a specific room or within a geographical area). Difficulties arise from the very nature of location information, which is dynamic, affected by a measurement error and requires a special dedicated infrastructure to be gathered. Rapid advancements in the field of wireless and mobile networking have fostered a new generation of devices suitable for being used as sensors by location technologies able to compute relative position and movement of users. Once a user's location has been gathered, an LBAC policy can be evaluated and the user could be granted access to

a particular resource. The location verification process must be able to tolerate rapid context changes, because mobile users can wander freely while initiating transactions by means of terminal devices like cell phones (GSM and 3G) and palmtops with wi-fi cards. Regardless to the specific technology, location verification can provide a rich context representation related to both users and resources they access. Location-based information possibly available to access control modules include the position and mobility of the requester when a certain access request is submitted. In the near future, location-based services are likely to provide a wealth of additional environment-related knowledge (e.g., is the user sitting at her desk or walking toward the door? Is she alone or together with others?). This kind of fine-grained context information potentially supports a new class of location-aware conditions regulating access to and fruition of resources.

1.4.1 Location-based predicates

The definition of location-based predicates for access control mechanisms requires to specify the conditions that an authorization language can support and today's location technology can verify. Three main classes of conditions could be identified [6]:

- *position-based* conditions on the location of a user, for evaluating, for example, whether a user is in a certain building or city or in the proximity of other entities;
- *movement-based* conditions on the mobility of a user, such as her velocity, acceleration, or direction where she is headed;
- *interaction-based* conditions relating multiple users or entities, for example, the number of users within a given area.

Although we have defined some specific predicates corresponding to specific conditions identified by the classes above, our language is extensible with respect to the predicates that can be added, as the need arises and technology progresses.

Furthermore, the language for location-based predicates assumes the following two elements.

- *users* is the set of *user identifiers* (UID) that unambiguously identify users known to the Location Services. This includes both users of the system (i.e., potential requesters) as well as any other known physical and/or moving entity which may need to be located (e.g., a vehicle with an on-board GPRS card). A typical UID for location-based applications is the SIM number linking the user's identity to a mobile terminal.[2]
- *areas* is a set of map regions identified either via a geometric model (i.e., a range in a n-dimensional coordinate space) or a symbolic model (i.e., with

[2]Individual users may carry multiple SIMs and the same SIMs may be passed over to other users. We shall not elaborate on these issues, since identity management in mobile networks is outside the scope of this chapter.

Table 1.1 Examples of location-based predicates

Type	Predicate	Description
Position	inarea(*user, area*)	Evaluate whether *user* is located within *area*.
	disjoint(*user*, *area*)	Evaluate whether *user* is located outside *area*.
	distance(*user*, *entity*, *min_dist*, *max_dist*)	Evaluate whether the distance between *user* and *entity* is within interval [*min_dist*, *max_dist*].
Movement	velocity(*user*, *min_vel*, *max_vel*)	Evaluate whether *user*'s speed falls within range [*min_vel*, *max_vel*].
Interaction	density(*area*, *min_num*, *max_num*)	Evaluate whether the number of users currently in *area* falls within interval [*min_num*, *max_num*].
	local_density(*user*, *area*, *min_num*, *max_num*)	Evaluate the density within a 'relative' area surrounding *user*.

reference to entities of the real world such as cells, streets, cities, zip code, buildings, and so on) [35].

In the following, we will refer to elements of *users* and of *areas* as *user* and *area terms*, respectively. While we assume such elements to be ground in the predicates, a language could be readily extended to support variables for them.

All predicates could be expressed as boolean queries, and therefore have the form *predicate*(*parameters, value*). Their evaluation returns a triple [*bool_value,* $\mathcal{R}$*, timeout*], where the term *bool_value* assumes values *True/False* according to the corresponding access decision, $\mathcal{R}$ represents a relevance value that qualifies the accuracy of the predicate evaluation, and *timeout* sets the validity timeframe of the location predicate evaluation. Our core set of location predicates includes the following predicates (see Table 1.1).

- A binary *position* predicate inarea whose first argument is a user term and second argument is an area term. The predicate evaluates whether a user is located within a specific area (e.g., a city, a street, a building).
- A binary *position* predicate disjoint whose first argument is a user term and second argument is an area term. The predicate evaluates whether a user is outside a specific area. Intuitively, disjoint is equivalent to the negation of inarea.
- A 4-ary *position* predicate distance whose first argument is a user term, second argument is either a user or area term (identifying an *entity* in the system), while the third and fourth arguments are two numbers specifying the minimum (*min_dist*) and maximum (*max_dist*) distance, respectively. The semantics of this predicate is to request whether the user lies within a given distance from the specified entity. The entity involved in the evaluation can be either stable or moving, physical or symbolic, and can be the resource to which the user is requesting access. Exact distance can be evaluated by setting the same value for *min_dist* and *max_dist*; "closer than" conditions can be evaluated by setting *min_dist* to 0; "farther than" conditions can be evaluated by setting *max_dist* to infinity.

- A ternary *movement* predicate velocity whose first argument is a user term, and the second and third arguments are two numbers specifying a minimum (*min_vel*) and maximum (*max_vel*) velocity, respectively. The semantics of the predicate is to request whether the user speed lies within a given range of velocity. Similarly to what happens for distance, exact velocity can be requested by setting the same value for *min_vel* and *max_vel*, while "smaller than" or "greater than" conditions can be evaluated by setting *min_vel* equal to 0 or *max_vel* equal to infinity, respectively.

- A ternary *interaction* predicate density whose first argument is an area term, while second and third arguments are numbers specifying a minimum (*min_num*) and maximum (*max_num*) number of users. The semantics of the predicate is to request whether the number of users currently in an *area* lies within the interval specified.

- A 4-ary *interaction* predicate local_density whose first argument is a user term, the second argument is a "relative" area with respect to the user, and the third and fourth arguments specify a minimum (*min_num*) and maximum (*max_num*) number of users, respectively. The semantics of the predicate is to evaluate the density within an area surrounding the user.

Example 1.4.1 *Let* `Alice` *be an element of* Users*, and* `Milan` *and* `Director Office` *be two elements of* Areas *(specifying two symbolic characterizations corresponding to two known ranges of spatial coordinates).*

inarea(`Alice`,`Milan`) = [`True`,`0.9`,`2007-08-09_11:10am`]
means that the Location Service assesses as true the fact that `Alice` *is located in* `Milan` *with a relevance* $\mathcal{R}$=`0.9`*, and that such an assessment is to be considered valid until* `11:10am` *of* `August 9, 2007`.

velocity(`Alice`,70,90) = [`True`,`0.7`,`2007-08-03_03:00pm`]
means that the Location Service assesses as true the fact that `Alice` *is traveling at a speed included in the range* [70,90] *with a relevance* $\mathcal{R}$=`0.7`*, and that such an assessment is to be considered valid until* `3:00pm` *of* `August 3, 2007`.

density(`Director Office`,0,1) = [`False`,`0.95`,`2005-08-21_06:00pm`]
means that the Location Service assesses as false the statement that there is at most one person in the `Director Office` *and believes that two or more persons are in the office with a relevance* $\mathcal{R}$=`0.95`*. Such an assessment is to be considered valid until* `06:00pm` *of* `August 21, 2007`.

1.4.2 Location-based access control policies

We now discuss how location-based access control policies can be expressed. Note that our goal is not to develop a new language for specifying access control policies. Instead, our proposal can be thought of as a general solution for enriching the

expressive power of existing languages (e.g., [8, 33, 51]), by exploiting location information, without increasing the computational complexity of their evaluation. We therefore assume that each user is assigned an identifier or pseudonym. Besides their identifiers/pseudonym, users usually have other properties (e.g., name, address, and date of birth) that can be transmitted through digital certificates and are grouped into a *user profile*. Objects are data/services which users may ask to access to. Properties of an object are grouped into an *object profile*. Each property into user or object profiles are referenced with the traditional dot notation. Also, to make it possible to refer to the user and object of the request being evaluated without introducing variables in the language, we rely on the **user** and **object** keywords. For instance, **user**.`Affiliation` indicates the property `Affiliation` within the profile of the user whose request is currently processed. A location-based authorization rule is the defined as follows.

Definition 1.4.1 (Location-based authorization rule) *A location-based authorization rule is a triple of the form* ⟨subject_expression, object_expression, actions⟩, *where:*

- subject_expression *is a boolean formula of terms that allows referring to a set of subjects depending on whether they satisfy or not certain conditions, where conditions can evaluate the user's profile, location predicates, or the user's membership in groups, active roles, and so on;*
- object_expression *is a boolean formula of terms that allows referring to a set of objects depending on whether they satisfy or not certain conditions, where conditions evaluate membership of the object in categories, values of properties on metadata, and so on;*
- actions *is the action (or set of actions) to which the policy refers.*

Conditions specified in the *subject_expression* field can be classified in two categories: *generic conditions* and *location-based conditions*. Generic conditions evaluate membership of subjects in classes or properties in their profiles and, as for the object expression, they are always of the form `predicate_name(`*arguments*`)`, where *arguments* is a list, possible empty, of constants or attributes. Location-based conditions are expressed with location predicates.

Example 1.4.2 *Let us consider a healthcare scenario where a hospital provides the Magnetic Resonance Imaging (MRI) examinations and is responsible for patients data management. Suppose that the MRI machine is the hardware/software that permits to do magnetic relevance tomography. Managing a MRI machine is a critical activity because privileges must be granted to strictly selected medical personnel only and must be performed according to high security standards (see* policy 1 *in Table 1.2). In addition, access to medical databases must be managed carefully and according to different security standards depending on the level of risk of the data to be accessed. In particular, access to examination data is critical, because they include high sensitive information about the state of health of hospital's customers*

Table 1.2 Examples of access control policies for a healthcare scenario

	subject_expression		actions	object_expression
	generic conditions	location conditions		
1	equal(**user**.Role, 'Doctor') ∧ Valid(**user**.Username, **user**.Password)	inarea(**user**.sim, MRI Control Room) ∧ density(MRI Room, 1, 1) ∧ velocity(**user**.sim, 0, 3)	Execute	equal(**object**.name, 'MRIMachine')
2	equal(**user**.Role, 'Doctor') ∧ Valid(**user**.Username, **user**.Password)	inarea(**user**.sim, Hospital) ∧ local_density(**user**.sim, Close By, 1, 1) ∧ velocity(**user**.sim, 0, 3)	Read	equal(**object**.category, 'Examination')
3	equal(**user**.Role, 'Nurse') ∧ Valid(**user**.Username, **user**.Password)	inarea(**user**.sim, First Aid) ∧ local_density(**user**.sim, Close By, 1, 1) ∧ velocity(**user**.sim, 0, 3)	Read	equal(**object**.category, 'Examination')
4	equal(**user**.Role, 'Doctor') ∧ Valid(**user**.Username, **user**.Password)	local_density(**user**.sim, Close By, 1, 1) ∧ disjoint(**user**.sim, Pharmaceutical Company)	Read	equal(**object**.category, 'Personal Info')
5	equal(**user**.Role, 'Secretary') ∧ Valid(**user**.Username, **user**.Password)	local_density(**user**.sim, Close By, 1, 1) ∧ inarea(**user**.sim, Hospital)	Read	equal(**object**.category, 'Log&Bill')

(see policies 2 and 3 *in Table 1.2). Patient-related information needs to be protected, for example, from disclosure to pharmaceutical companies (see* policy 4 *in Table 1.2). Finally, access to logging and billing data of the patients are usually less critical but still to be handled in a highly secured environment and to be granted only to selected personnel, according to the laws and regulations in force (see* policy 5 *in Table 1.2).*

1.5 OBFUSCATION TECHNIQUES FOR USER-PRIVACY

To guarantee user location privacy, we introduce three basic obfuscation techniques that modify a user location to reduce the associated relevance (henceforth the accuracy) until a given level.

1.5.1 Obfuscation by enlarging the radius

Obfuscating a location measurement area by increasing its radius (see Figure 1.3(a)) is the technique that most solutions exploit, either explicitly or implicitly by scaling a location to a coarser granularity (e.g., from meters to hundred of meters, from a city block to the whole town, and so on). The obfuscation is a probabilistic effect due to a decreasing of the corresponding joint probability density function (joint pdf), which can be expressed as $\forall r, r', r < r' : f_r(x, y) > f_{r'}(x, y)$. The following proposition allows us to calculate the obfuscated area.

Proposition 1 *Given a location area of radius r with relevance $\mathcal{R}_{Init}$, and an obfuscated area of radius r' derived by enlarging the original radius, the relevance $\mathcal{R}_{Final}$ of the obfuscated area is calculated by reducing $\mathcal{R}_{Init}$ of the ratio $\frac{f_{r'}(x,y)}{f_r(x,y)}$ of corresponding joint pdf.*

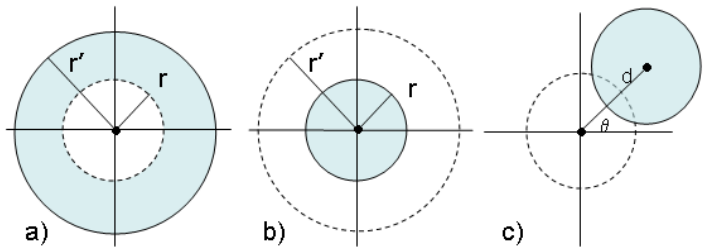

Fig. 1.3 Obfuscation by enlarging the radius (a), reducing the radius (b) and shifting the center (c)

From the assumption of uniform distribution over a circular area, the relation between $\mathcal{R}_{Final}$ and $\mathcal{R}_{Init}$ can be written as:

$$\mathcal{R}_{Final} = \frac{f_{r'}(x,y)}{f_r(x,y)}\mathcal{R}_{Init} = \frac{\frac{1}{\pi r'^2}}{\frac{1}{\pi r^2}}\mathcal{R}_{Init} = \frac{r^2}{r'^2}\mathcal{R}_{Init} \begin{cases} = \mathcal{R}_{Init} & r' = r \\ \in (\mathcal{R}_{Init}, 0) & r' > r \\ = 0 & r' \to +\infty \end{cases} \tag{1.3}$$

Therefore, given the two relevances $\mathcal{R}_{Init}$ and $\mathcal{R}_{Final}$, and the radius r of the initial area, an obfuscated area calculated with this technique has a final radius: $r' = r\sqrt{\frac{\mathcal{R}_{Init}}{\mathcal{R}_{Final}}}$.

1.5.2 Obfuscation by reducing the radius

Another possible way of obfuscating a user location consists in reducing the radius r of one location to a smaller r', as showed in Figure 1.3(b). The obfuscation effect, in this case, is produced by a correspondent reduction of the probability to find the real user location within the returned area, while the joint pdf is fixed.

To state it formally, let us consider the unknown real user position coordinates (x_u, y_u). Given a location area of radius r, the probability that the real user position falls in the area is $P((x_u, y_u) \in Area(r, x, y))$. When we obfuscate by reducing the radius, an area of radius $r' \leq r$ is returned, which implies that $P((x_u, y_u) \in Area(r', x, y)) \leq P((x_u, y_u) \in Area(r, x, y))$, because a circular ring having pdf greater than zero has been excluded.

The obfuscated area, derived by a radius reduction, can be calculated by considering the following proposition.

Proposition 2 *Given a location measurement area of radius r with relevance $\mathcal{R}_{Init}$, and an obfuscated area of radius r' derived by reducing the radius, relevance $\mathcal{R}_{Final}$ of the obfuscated area is calculated by reducing $\mathcal{R}_{Init}$ of the ratio $\frac{P((x_u,y_u)\in Area(r',x,y))}{P((x_u,y_u)\in Area(r,x,y))}$ of corresponding probabilities.*

In cartesian coordinates, $P((x,y) \in A)$, for all subsets $A \subseteq \mathbb{R}^2$, is calculated as $\iint_A f(x,y)dxdy$, being $f(x,y)$ the corresponding joint pdf. Changing to polar coordinates (s,θ) and solving the double integral requires the transformation $(x,y) \rightarrow (s,\theta)$, which gives $dxdy = sdsd\theta$ [43]. The pdf, instead, remains unchanged to the value obtained from the original location measurement, i.e., $f(r,\theta) = \frac{1}{\pi r^2}$. According to these observations, we have that:

$$P((x_u,y_u) \in Area(r',x,y)) = \int_0^{2\pi}\int_0^{r'} f(r,\theta)sdsd\theta = 2\pi\int_0^{r'}\frac{s}{\pi r^2}ds = \frac{2}{r^2}\int_0^{r'} sds = \frac{r'^2}{r^2}$$

Analogously, $P((x_u,y_u) \in Area(r,x,y))$ can be calculated as

$$P((x_u,y_u) \in Area(r,x,y)) = \int_0^{2\pi}\int_0^{r'} f(r,\theta)sdsd\theta + \int_0^{2\pi}\int_{r'}^{r} f(r,\theta)sdsd\theta = 1$$

resulting in a probability equals to 1 of having the user u inside the location measurement $Area(r,x,y)$. Therefore, the relation stated in the proposition can be written as:

$$\mathcal{R}_{Final} = \frac{Pr((x_u,y_u)\in Area(r',x_c,y_c))}{Pr((x_u,y_u)\in Area(r,x_c,y_c))}\mathcal{R}_{Init} = \frac{r'^2}{r^2}\mathcal{R}_{Init}\begin{cases} = \mathcal{R}_{Init} & r' = r \\ \in (\mathcal{R}_{Init},0) & r' < r \\ = 0 & r' \rightarrow 0 \end{cases} \quad (1.4)$$

Therefore, given the two relevances $\mathcal{R}_{Init}$ and $\mathcal{R}_{Final}$, and the radius r of the initial area, an obfuscated area calculated with this technique has a final radius: $r' = r\sqrt{\frac{\mathcal{R}_{Final}}{\mathcal{R}_{Init}}}$.

1.5.3 Obfuscation by shifting the center

Location obfuscation can also be achieved by shifting the center of the location measurement area and returning the displaced area, as showed in Figure 1.3(c). Intuitively, the obfuscation effect depends on the intersection of the two areas, i.e., the smaller the intersection, the highest the obfuscation. In this case, it should be considered unacceptable to produce obfuscated areas disjoint from the original measured location area. The reason is that all disjoint areas would have probability equal to zero of including the real user location, and then they would be indistinguishable for our relevance estimator. Such cases are considered as just false location information, which, by design, our system does not produce, assuming that LBS and related applications such as LBAC [6, 7] cannot, in general, deal with false information in the provision of a business service.

We call d the distance between the centers and r the radius of the areas, here assumed to be the same and, since the original and the obfuscated areas cannot be disjoint, $d \in [0,2r]$. In particular, if d=0, there is no privacy gain; if d=$2r$, there is maximum privacy; and if $0 < d < 2r$, there is an increment of privacy. In addition to distance d, a rotation angle θ must be specified to derive an obfuscated area by shifting the center. For the scope of this paper, angle θ can be assumed to

be generated randomly with no loss of generality. Strategies for selecting a value of angle θ depends on the application context and are deeply analyzed in [3, 7].

Given d and θ, we denote the obfuscated area as $Area(r, x+d\sin\theta, y+d\cos\theta)$ and the intersection between the original and the obfuscated area as $Area_{Init\cap Final} = Area(r,x,y) \bigcap Area(r, x+d\sin\theta, y+d\cos\theta)$.

To measure the obfuscation effect and define the relation between relevances, two probabilities must be composed. The first is the probability that the real user position falls in the intersection $Area_{Init\cap Final}$, i.e., $P((x_u,y_u) \in Area_{Init\cap Final}|(x_u,y_u) \in Area(r,x,y))$. The second is the probability that one point selected from the whole obfuscated area belongs to the intersection, i.e., $P((x',y') \in Area_{Init\cap Final}|(x',y') \in Area(r, x+d\sin\theta, y+d\cos\theta))$. The product of these two probabilities estimates the reduction of the relevance due to the obfuscation. The obfuscated area, derived by shifting the center, can be calculated by considering the following proposition.

Proposition 3 *Given a measured location area of radius $r = r_{meas}$ with initial relevance $\mathcal{R}_{Init}$, and an obfuscated area of same radius derived by shifting the original center of distance d and angle θ, $\mathcal{R}_{Final}$ is calculated by multiplying $\mathcal{R}_{Init}$ by:*

$$Pr((x_u,y_u) \in Area_{Init\cap Final}|(x_u,y_u) \in Area(r,x,y)) \cdot Pr((x',y') \in Area_{Init\cap Final}|(x',y') \in Area(r, x+d\sin\theta, y+d\cos\theta)).$$

Since the two probabilities can then be expressed as:

$$Pr((x_u,y_u) \in Area_{Init\cap Final}|(x_u,y_u) \in Area(r,x,y)) = \frac{Area_{Init\cap Final}}{Area(r,x,y)}$$
$$Pr((x',y') \in Area_{Init\cap Final}|(x',y') \in Area(r, x+d\sin\theta, y+d\cos\theta)) = \frac{Area_{Init\cap Final}}{Area(r, x+d\sin\theta, y+d\cos\theta)}$$

it follows that:

$$\mathcal{R}_{Final} = \frac{Area_{Init\cap Final} \cdot Area_{Init\cap Final}}{Area(r,x,y) \cdot Area(r, x+d\sin\theta, y+d\cos\theta)} \mathcal{R}_{Init} \begin{cases} = \mathcal{R}_{Init} & d = 0 \\ \in (\mathcal{R}_{Init}, 0) & 0 < d < 2r \\ = 0 & d = 2r \end{cases} \tag{1.5}$$

Expanding the term $Area_{Init\cap Final}$ as a function of distance d between the centers, distance d can be calculated numerically by solving the following system of equations whose variables are d and σ.

$$\begin{cases} \sigma - \sin\sigma = \sqrt{\delta}\pi \quad with\ \delta = \frac{Area_{Init\cap Final} \cdot Area_{Init\cap Final}}{Area(r,x,y) \cdot Area(r, x+d\sin\theta, y+d\cos\theta)} \\ d = 2r\cos\frac{\sigma}{2} \end{cases} \tag{1.6}$$

Variable σ represents the central angle of the circular sector identified by the two radii connecting the center of the original area with the intersection points of the original and the obfuscated areas. These two equations represent the solution to the problem of calculating the distance d between the centers of two partially overlapped circumferences, in the special case of same radius.

1.6 A PRIVACY-AWARE LBAC SYSTEM

We now present a privacy-aware LBAC system that integrates the obfuscation techniques with the location-based access control system previously described.

1.6.1 LBAC predicates evaluation: $\mathcal{R}_{Eval}$ calculation

A major design issue for a privacy-aware LBAC architecture is related to the component in charge of evaluating LBAC predicates. Two choices are possible, which deeply affect how privacy is guaranteed.

- *ACE evaluation*: ACE asks users locations to LM without disclosing LBAC predicates. Locations are returned together with a relevance value.
- *LM evaluation*: ACE sends to LM a LBAC predicate for evaluation and receives a boolean answer and a relevance value.

Both choices are viable and well-suited for different set of requirements. On one side, *ACE evaluation* enforces a clear separation between applications and location services because the location service infrastructure (i.e., LM and LPs) never deals with application-dependent location-based predicates. On the other side, *LM evaluation* avoids the exchange of user locations, although obfuscated, with applications. This second choice is also more flexible in business terms. For instance, an ACE can subscribe to a location service for a specific set of location predicates, and select different QoS according to different needs (e.g., different accuracy levels). The LM could then differentiate prices according to service quality. Since more elaborate, in the following we focus on this second option.

As previously discussed, we assume that the results returned by LM have the form (*bool_value*,$\mathcal{R}$,*timeout*). While in the case of ACE evaluation, relevance $\mathcal{R}$ contained into the response is the $\mathcal{R}_{Final}$ value obtained by obfuscating a measured location, in case of LM evaluation, value $\mathcal{R}$ is the result of an additional elaboration that depends on the type of location predicate. It is important to highlight that *movement* and *interaction* predicates are intrinsically different from *position* predicates since they do not release users location bound to identities, and their evaluation involves more than the location measurement of a user and the area specified in the location predicate. For the sake of clarity, we call henceforth $\mathcal{R}_{Eval}$ the parameter $\mathcal{R}$ contained into a response produced after a LM evaluation.

Position predicates. Suppose that a location measurement $Area(r_{meas}, x_c, y_c)$ ($Area_{Init}$ below) with relevance $\mathcal{R}_{Init}$ has been obfuscated producing an area $Area_{Final}$ with relevance $\mathcal{R}_{Final}$. Relevance $\mathcal{R}_{Eval}$ is derived from $\mathcal{R}_{Final}$ by considering both the obfuscated area and the area specified in the LBAC predicate. LM calculates $\mathcal{R}_{Eval}$ of the predicate evaluation as follows:

$$\mathcal{R}_{Eval} = \frac{Area_{Final \cap LBAC}}{Area_{Final}} \cdot \mathcal{R}_{Final}$$

where the scalar factor depends on the intersection, denoted $Area_{Final \cap LBAC}$, between the obfuscated area and the area specified by the LBAC predicate. Suppose that `inarea`(*John*, *Room1*) is the predicate that the ACE component sends to the LM component, which asks whether the user *John* is in room *Room1*. If John's position has an overlap greater than zero with *Room1*, the predicate evaluation returns (`true`,$\mathcal{R}_{Eval}$,*timeout*). Otherwise, for all positions disjoint with *Room1*, the evaluation returns (`true`,$\mathcal{R}_{Eval} \rightarrow 0$,*timeout*).

Movement predicates. Predicate velocity, the only predicate that we currently have defined, is evaluated by first measuring two user positions at different times, and then by calculating her velocity. Relevance $\mathcal{R}_{Eval}$ cannot be generated as for the previous case. Rather, it is generated by considering the mean value of relevances $\mathcal{R}_{Init}$ associated with the positions used to calculate user's velocity.

$$\mathcal{R}_{Eval} = \frac{\mathcal{R}_{Init_1} + \mathcal{R}_{Init_2}}{2}$$

Although estimating a user's velocity does not release information about user location, the user can choose to obfuscate the velocity result. In this case, $\mathcal{R}_{Eval}$ is calculated as the mean of relevances $\mathcal{R}_{Final}$ associated with the obfuscated positions used to calculate the velocity of the user.

$$\mathcal{R}_{Eval} = \frac{\mathcal{R}_{Final_1} + \mathcal{R}_{Final_2}}{2}$$

Suppose that `velocity`(*John*, *70*, *90*) is the predicate that the ACE component sends to the LM component, which asks whether velocity of user *John* is within the range [70,90]. If John's velocity is in the specified interval, the predicate evaluation returns (`true`,$\mathcal{R}_{Eval}$,*timeout*). Otherwise, the evaluation returns (`true`,$\mathcal{R}_{Eval} \rightarrow 0$,*timeout*).

Interaction predicates. Relevance $\mathcal{R}_{Eval}$ is calculated by using location measurements of all users locations intersecting a reference area called $Area_{LBAC}$. Two predicates have been defined. The density predicate, which requires that $Area_{LBAC}$ is geographically identified (e.g., a city), and predicate local_density, which, instead, considers as $Area_{LBAC}$ a given area around a user. $\mathcal{R}_{Eval}$ is calculated from equation (1.7) as follows:

$$\mathcal{R}_{Eval} = \frac{\sum_{i=1}^{n} \frac{Area_{i,Init \cap LBAC}}{Area_{i,Init}} \cdot \mathcal{R}_{Init\,i}}{n} \quad \forall Area_{i,Init} : Area_{i,Init \cap LBAC} \neq 0$$

where $Area_{i,Init \cap LBAC}$ represents the intersection between the *i*-th location measurement $Area_{i,Init}$ and the area identified by the LBAC predicate, $\mathcal{R}_{Init\,i}$ is the initial relevance of the *i*-th location measurement, and n is the number of users involved in the predicate evaluation. Whether obfuscated areas are considered, i.e., all the areas generated by location measurements of users that have an intersection with $Area_{LBAC}$, $\mathcal{R}_{Eval}$ is calculated starting from equation (1.7) as follow:

$$\mathcal{R}_{Eval} = \frac{\sum_{i=1}^{n} \frac{Area_{i,Final \cap LBAC}}{Area_{i,Final}} \cdot \mathcal{R}_{Final\,i}}{n}$$

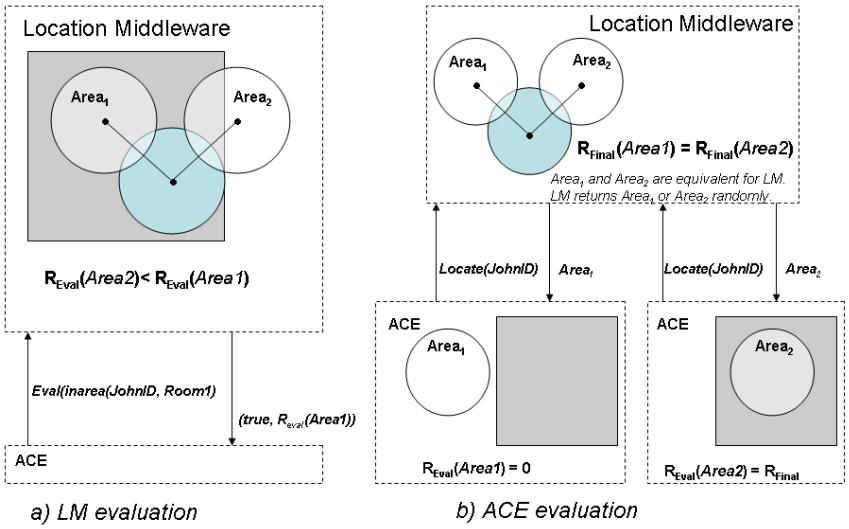

Fig. 1.4 An example of LM evaluation (a), and ACE evaluation (b)

Suppose that `density`(*Room1*, *0*, *3*) is the predicate that the ACE component sends to the LM component, which asks whether the number of users in *Room1* is between 0 and 3. If the number of users is in the interval, the predicate evaluation returns (`true`,$\mathcal{R}_{Eval}$,*timeout*). Otherwise, the evaluation returns (`true`,$\mathcal{R}_{Eval}\rightarrow$0,*timeout*).

1.6.1.1 LM vs ACE evaluation To further analyze the differences between the adoption of ACE or LM evaluation, we focus on a scenario where an obfuscation by shifting the center is applied and a position predicate is evaluated.[3] In this case, the ACE vs LM choice has a significant impact. Consider the examples in Fig. 1.4(a) and Fig. 1.4(b) that show the evaluation of predicate `inarea`(*John*, *Room1*) in case of LM evaluation and of ACE evaluation, respectively. When obfuscation by shifting the center is applied, there are infinite values of angle θ that could be chosen, all equivalent with respect to the relevance value $\mathcal{R}_{Final}$. Here, *Area1* and *Area2* are two possible obfuscated areas.

If LM evaluation is performed, LM computes $\mathcal{R}_{Eval}$, as previously seen, and is able to establish an ordering among obfuscated areas according to the different values of $\mathcal{R}_{Eval}$. In our example, it is easy to see that relevance $\mathcal{R}_{Eval}$resulting from *Area1*, denoted as $\mathcal{R}_{Eval}$*(Area1)*, is greater than relevance $\mathcal{R}_{Eval}$resulting from *Area2*, denoted as $\mathcal{R}_{Eval}$*(Area2)*. This information is important for the provision of the location service, because when returned to ACE, the value $\mathcal{R}_{Eval}$ is matched with $\mathcal{R}_{LBAC}$, the minimum relevance required by ACE for LBAC evaluation. The best

[3] Same discussion holds for movement and interaction predicates.

strategy for LM is therefore to select the angle θ that produces the obfuscated area that, given $\mathcal{R}_{Final}$, maximizes $\mathcal{R}_{Eval}$.[4]

If ACE evaluation is in place, LM does not calculate any $\mathcal{R}_{Eval}$ (i.e., $\mathcal{R}_{Eval}$ is just equal to $\mathcal{R}_{Final}$), and it can only select randomly one value for θ among all those that produce an obfuscated area with same $\mathcal{R}_{Final}$. In this way, random selection of the obfuscated area (in our example, *Area1* or *Area2*) may cause an unpredictable result during ACE evaluation, ranging from relevance equal to zero (e.g., when *Area1* in Fig. 1.4(b) is returned) to relevance equal to $\mathcal{R}_{Final}$ (e.g., when *Area2* in Fig. 1.4(b) is returned). As a consequence, also the matching with the condition over $\mathcal{R}_{LBAC}$ results in random rejection or acceptance of the predicate evaluation. Therefore, obfuscation by shifting the center is incompatible with ACE evaluation. This result supports architectures including location middleware capable of autonomously evaluating LBAC predicates.

Finally, there is a subtlety to consider when obfuscation by shifting the center is applied. When a LBAC predicate is evaluated the choice of θ is relevant, because according to the position of the obfuscated area, the value of $\mathcal{R}_{Eval}$ may change. Therefore LM could try to select the θ angle that maximizes $\mathcal{R}_{Eval}$. Fig. 1.5 shows an example with three obfuscated areas, namely *Area1*, *Area2*, and *Area3*, which provide the same $\mathcal{R}_{Final}$ value and different $\mathcal{R}_{Eval}$ values, denoted $\mathcal{R}_{Eval}$(*Area1*), $\mathcal{R}_{Eval}$(*Area2*), and $\mathcal{R}_{Eval}$(*Area3*), respectively. It is easy to see that $\mathcal{R}_{Eval}$(*Area1*) is greater than $\mathcal{R}_{Eval}$(*Area2*) (i.e., the overlap between *Area1* and *Milan* is larger than the overlap between *Area2* and *Milan*) and, correspondingly, the value of angle θ that LM should take into consideration is the one that produces *Area1*.

A problem could arises with *Area3*, which has clearly the greatest overlap with *Milan*. *Area3* could provide a $\mathcal{R}_{Eval}$ greater than the one that would have provided the original area $Area_{Init}$. This would lead to an inconsistent LBAC predicate evaluation. The reason is that LM would have an incentive to configure obfuscation as a way to artificially increase the odds of satisfying the $\mathcal{R}_{LBAC}$ threshold. To avoid such a side effect, we introduce the following additional constraint: *relevance $\mathcal{R}_{Eval}$ derived from the obfuscated area with relevance $\mathcal{R}_{Final}$ must be lesser than or equal to the one provided by the original area with relevance $\mathcal{R}_{Init}$, that is $\mathcal{R}_{Eval}(Area_{Final}) \leq \mathcal{R}_{Eval}(Area_{Init})$.* In other terms, areas must not be manipulated with obfuscation techniques just to increase the odds of satisfying LBAC quality requirements. Our constraint ensures that, given an infinite set Θ of angles, a set $\Theta_f \subseteq \Theta$ is generated, containing all the valid angles $\theta_1 \dots \theta_n$ that produce a relevance $\mathcal{R}_{Eval}$ at most equals to the relevance produced by considering the original area.

In the example of inarea evaluation in Fig. 1.5, the following restriction is introduced,

$$\mathcal{R}_{Eval} \leq \frac{Area_{Init \cap LBAC}}{Area_{Init}} \cdot \mathcal{R}_{Init} \tag{1.7}$$

[4]In addition to the strategy that selects the θ angle that maximizes $\mathcal{R}_{Eval}$, other strategies can be exploited for selecting θ (e.g., a random choice).

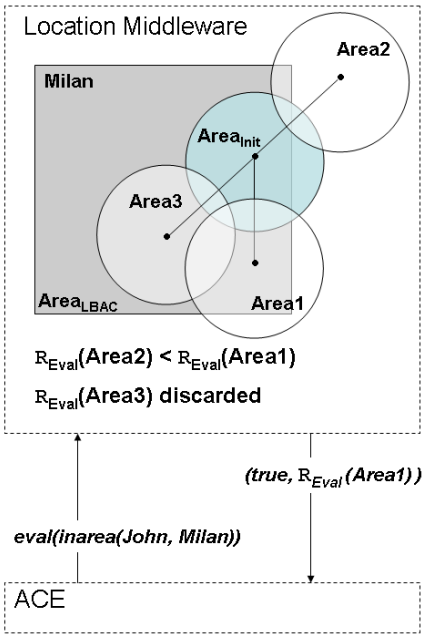

Fig. 1.5 Area selection

and *Area3*, which does not satisfy this constraint, is discarded in favour of *Area1*.

1.6.2 The privacy-aware middleware

Currently available middleware components are mostly in charge of managing interactions between applications and location providers, and managing communication and negotiation protocols aimed at maximizing the QoS [30, 38, 40, 41, 47]. In a privacy-aware LBAC, a middleware component is also responsible for balancing users privacy and location-based services accuracy. To this end, our LM provides functionalities for both the obfuscation of users locations and the location-based predicates evaluation. As shown in Fig. 1.6, LM is functionally divided into the following five logical components.

- *Communication Layer.* It manages the communication process with LPs. It hides low-level communication details to other components.
- *Negotiation Manager.* It acts as an interface with ACE for negotiating QoS attributes [4].
- *Access Control Preference Manager.* It manages location service attributes by interacting with the Location Obfuscation component.

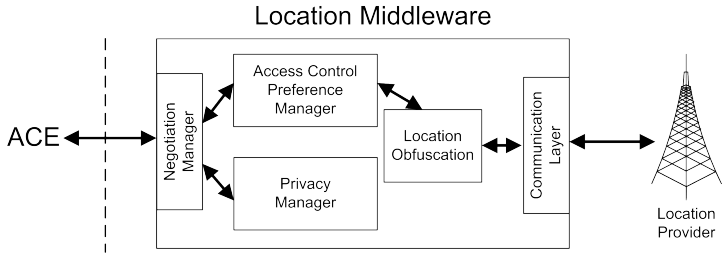

Fig. 1.6 Location Middleware

- *Location Obfuscation.* It applies obfuscation techniques for users privacy.
- *Privacy Manager.* It manages privacy preferences and location-based predicate evaluation.

It is important to highlight that the architecture of our location middleware can be extended to include the important case of users setting *multiple privacy preferences* according to different contexts. For instance, there could be users wishing to set: *i)* no privacy preferences for location services dedicated to the social network of their relatives and close friends; *ii)* a certain level of privacy for business location services aimed at helping to find point of interests (e.g., shops, or monuments), and for location services whose goal is to find their position while at work; and *iii)* strong privacy requirements in high sensitive contexts.

To conclude, we provide two examples of LBAC predicates evaluation based on two location predicates: `inarea` and `distance`. Specifically, `inarea(`*user_term*,*area_term*`)` evaluates whether *user_term* is located within *area_term*, and `distance(`*user*,*entity*,d_{min},d_{max}`)` evaluates whether the distance between *user* and *entity* is within the interval [d_{min},d_{max}].

Example 1.6.1 *Suppose that ACE requires users (*John *in this example) to be located in* Milan *with a relevance $\mathcal{R}_{LBAC}$=0.5 to access a service. Also, suppose that* John*'s privacy preference requires a relevance $\mathcal{R}_{Final}$=0.8. To enforce John's access request, the ACE asks the LM to evaluate the predicate* `inarea(`John, Milan`)`*, where* John *represents the located user. Let the location measurement of* John *be $Area_{Init}$ with $\mathcal{R}_{Init}$=1. Fig. 1.7 shows graphically an example of $\mathcal{R}_{Eval}$ computation when the obfuscation by enlarging the radius is applied. The scalar factor $\frac{Area_{Final \cap LBAC}}{Area_{Final}}$ is equal to 0.75. From (1.7), we can produce the final relevance $\mathcal{R}_{Eval}$ associated with the predicate evaluation: $\mathcal{R}_{Eval}$=$0.75 \cdot \mathcal{R}_{Final}$=0.6. The predicate evaluation process is concluded and the result $(True, 0.6, timeout)$ is returned to the ACE. Finally, the ACE compares $\mathcal{R}_{Eval}$ with $\mathcal{R}_{LBAC}$. Since $\mathcal{R}_{LBAC} < \mathcal{R}_{Eval}$, the quality of the evaluation satisfies the ACE requirements, and* John *gains the access.*

Example 1.6.2 *Suppose that the ACE requires users (again* John *in this example) to stay at least 1000m away from the* Dangerous *area of Fig. 1.8 used for stocking dangerous material with a relevance $\mathcal{R}_{LBAC}$=0.8 to access a service. Then,*

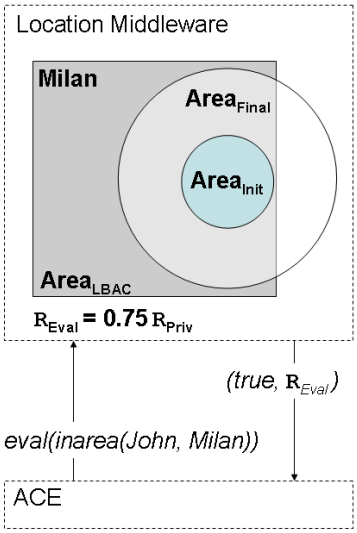

Fig. 1.7 LM `inarea` predicate evaluation

suppose that John's privacy preference requires a relevance $\mathcal{R}_{Final}$*=0.2. Whenever* John *submits an access request, the ACE asks the LM to evaluate the predicate* `distance`(John,Dangerous,d_{min},d_{max})*, where* John *represents the located user,* d_{min}*=1000m, and* $d_{max} = +\infty$*. The predicate distance identifies an area* $Area_{LBAC}$ *(see grey area in Fig. 1.8), around the* Dangerous *area, which contains all the points outside the* Dangerous *area that have a distance between* d_{min} *and* d_{max}*. Let the location measurement of John be* $Area_{Init}$ *with* $\mathcal{R}_{Init}$*=0.9. Fig. 1.8 shows graphically an example of* $\mathcal{R}_{Eval}$ *computation when the obfuscation by shifting the center is applied. Since the intersection between the obfuscated area* $Area_{Final}$ *and* $Area_{LBAC}$ *is equal to half of the* $Area_{Final}$*, the scalar factor* $\frac{Area_{Final \cap LBAC}}{Area_{Final}}$ *is equal to 0.5. From (1.7), we calculate the final relevance* $\mathcal{R}_{Eval}$ *associated with the predicate evaluation:* $\mathcal{R}_{Eval}$*=*$\frac{Area_{Final \cap LBAC}}{Area_{Final}} \cdot \mathcal{R}_{Final}$*=0.1. The predicate evaluation process is concluded and the result* $(True, 0.1, timeout)$ *is returned to the ACE meaning that* John *is far from the* Dangerous *area of at least* d_{min} *with a relevance of 0.1. Finally, since* $\mathcal{R}_{LBAC} > \mathcal{R}_{Eval}$*, the ACE denies John's request.*

1.7 CONCLUSIONS

This chapter has discussed requirements for the design of location-based access control systems and their main differences with respect to traditional access control solutions. Details have been provided for extending an authorization language, and for the evaluation and enforcement of location conditions. Privacy requirements for protecting location information have also been described. In particular, the trade-

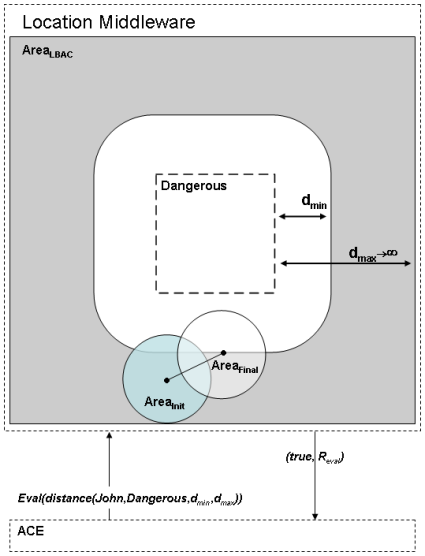

Fig. 1.8 LM `distance` predicate evaluation

off between the need of information accuracy required by LBAC systems and the obfuscation of the same information for privacy reasons has been considered. Some basic obfuscation techniques have been defined together with a general metric, called relevance, that can be used for both measuring the degree of location privacy and the degree of accuracy required. Examples and case studies enriched the presentation of issues and concepts. There are, however, many research issues that need to be further investigated, such as the analysis of secondary effects of location predicate evaluation, de-obfuscation attacks, and strategies for the negotiation of QoS attributes.

Acknowledgments

This work was partially supported by the European Union within the 6FP project PRIME under contract IST-2002-507591, by the European Union within the 7FP project "PrimeLife", and by the Italian MIUR within PRIN 2006 under project 2006099978.

REFERENCES

1. I.F. Akyildiz and J.S.M. Ho. Dynamic mobile user location update for wireless PCS networks. *Wireless Networks*, 1995.

2. M. Anisetti, C.A. Ardagna, V. Bellandi, and E. Damiani. Positioning method and system for mobile communications networks, related networks and computer program product. In *European Patent No. 05425643.3*, Deposited in date 15 September 2005, Patent Pending.

3. C. A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. A middleware architecture for integrating privacy preferences and location accuracy. In *Proc. of IFIP SEC 2007 Conference*, Sandton, Gauteng, South Africa, May 2007.

4. C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. Location-based metadata and negotiation protocols for lbac in a one-to-many scenario. In *Proc. of the Workshop on Security and Privacy in Mobile and Wireless Networking (SecPri_MobiWi 2006)*, Coimbra, Portugal, May 2006.

5. C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and S. Samarati. Location privacy protection through obfuscation-based techniques. In *Proc. of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Redondo Beach, CA, USA, July 2007.

6. C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. Supporting location-based conditions in access control policies. In *Proc. of the ACM Symposium on Information, Computer and Communications Security (ASIACCS'06)*, Taipei, Taiwan, March 2006.

7. C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. Managing privacy in LBAC systems. In *Proc. of the Second IEEE International Symposium on Pervasive Computing and Ad Hoc Communications (PCAC-07)*, Niagara Falls, Canada, May 2007.

8. C.A. Ardagna, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. Towards Privacy-Enhanced Authorization Policies and Languages. In *Proc. of the 19th IFIP WG11.3 Working Conference on Data and Application Security*, Nathan Hale Inn, University of Connecticut, Storrs, USA, August 2005.

9. P. Bellavista, A. Corradi, and C. Giannelli. Efficiently managing location information with privacy requirements in wi-fi networks: a middleware approach. In *Proc. of the International Symposium on Wireless Communication Systems (ISWCS'05)*, Siena, Italy, September 2005.

10. A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, January-March 2003.

11. A. R. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *Proc. of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW04)*, Orlando, FL, March 2004.

12. C. Bettini, X.S. Wang, and S. Jajodia. Protecting privacy against location-based personal identification. In *Proc. of the 2nd VLDB Workshop on Secure Data Management*, Trondheim, Norway, September 2005.

13. J. Borkowski, J. Niemelä, and J. Lempiäinen. Performance of cell id+rtt hybrid positioning method for umts radio networks. In *Proc. of the 5th European Wireless Conference*, Barcelona, Spain, February 2004.

14. L. Cong and W. Zhuang. Hybrid TDOA/AOA mobile user location for wideband CDMA cellular systems. *IEEE Transactions on Wireless Communications*, 1(5):439–447, July 2002.

15. M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proc. of the 3rd International Conference PERVASIVE 2005*, Munich, Germany, May 2005.

16. M. Duckham and L. Kulik. Simulation of obfuscation and negotiation for location privacy. In *Proc. of COSIT 2005*, pages 31–48, Ellicottville, NY, September 2005.

17. D. Faria and D. Cheriton. No long-term secrets: Location-based security in overprovisioned wireless lans. In *Proc. of the 3rd ACM Workshop on Hot Topics in Networks (HotNets-III)*, San Diego, CA, November 2004.

18. S. Garg, M. Kappes, and M. Mani. Wireless access server for quality of service and location based access control in 802.11 networks. In *Proc. of the 7th IEEE Symposium on Computers and Communications (ISCC 2002)*, Taormina/Giardini Naxos, Italy, July 2002.

19. B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *Proc. of the 25th International Conference on Distributed Computing Systems (IEEE ICDCS 2005)*, Columbus, OH, June 2005.

20. Geographic Location/Privacy (geopriv). September 2006. http://www.ietf.org/html.charters/geopriv-charter.html.

21. I. Getting. The global positioning system. *IEEE Spectrum*, 30(12):36–47, December 1993.

22. M. Gruteser, J. Bredin, and D. Grunwald. Path privacy in location-aware computing. In *Proc. of the Second International Conference on Mobile Systems, Application and Services (MobiSys2004)*, Boston, MA, June 2004.

23. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of the 1st International Conference on Mobile Systems, Applications, and Services*, San Francisco, CA, May 2003.

24. M. Gruteser and Xuan Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security & Privacy Magazine*, 2(2):28–34, March-April 2004.

25. F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks: Possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Processing Magazine*, 22(4):41–53, July 2005.

26. C. Hauser and M. Kabatnik. Towards Privacy Support in a Global Location Service. In *Proc. of the IFIP Workshop on IP and ATM Traffic Management (WATM/EUNICE 2001)*, Paris, France, September 2001.

27. U. Hengartner and P. Steenkiste. Protecting access to people location information. *Proc. of the First International Conference on Security in Pervasive Computing Security in Pervasive Computing*, Boppard, Germany, March 2003.

28. U. Hengartner and P. Steenkiste. Implementing access control to people location information. In *Proc. of the 9th ACM Symposium on Access Control Models and Technologies 2004 (SACMAT 2004)*, Yorktown Heights, NY, June 2004.

29. B. Ho and M. Gruteser. Protecting location privacy through path confusion. In *Proc. of IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)*, Athens, Greece, September 2005.

30. D. Hong, M. Yuan, and V. Y. Shen. Dynamic privacy management: a plug-in service for the middleware in pervasive computing. In *Proc. of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services (MobileHCI'05)*, Salzburg, Austria, September 2005.

31. S. Horsmanheimo, H. Jormakka, and J. Lahteenmaki. Location-aided planning in mobile network–Trial results. *Wireless Personal Communications: An International Journal*, 30(2-4):207–216, September 2004.

32. H. Hu and D.L. Lee. Energy-efficient monitoring of spatial predicates over moving objects. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 28(3):19–26, September 2005.

33. S. Jajodia, P. Samarati, M.L. Sapino, and V.S. Subrahmanian. Flexible support for multiple access control policies. *ACM Transactions on Database Systems*, 26(2):214–260, June 2001.

34. M. Langheinrich. A privacy awareness system for ubiquitous computing environments. In *Proc. of the 4th International Conference on Ubiquitous Computing (Ubicomp 2002)*, Göteborg, Sweden, September 2002.

35. N. Marsit, A. Hameurlain, Z. Mammeri, and F. Morvan. Query processing in mobile environments: a survey and open problems. In *Proc. of the 1st*

International Conference on Distributed Framework for Multimedia Applications (DFMA'05), Besancon, France, February 2005.

36. M.F. Mokbel and W.G. Aref. GPAC: generic and progressive processing of mobile queries over mobile data. In *Proc. of the 6th International Conference on Mobile data management*, Ayia Napa, Cyprus, May 2005.

37. M. Mokbel, C.-Y. Chow, and W. Aref. The new Casper: Query processing for location services without compromising privacy. In *Proc. of the 32nd International Conference on Very Large Data Bases*, Seoul, Korea, September 2006.

38. G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2(1):56–64, January-March 2003.

39. J. Myllymaki and S. Edlund. Location aggregation from multiple sources. In *Proc. of the 3rd IEEE International Conference on Mobile Data Management (MDM 02)*, Singapore, January 2002.

40. H. Naguib, G. Coulouris, and S. Mitchell. Middleware support for context-aware multimedia applications. In *Proc. of the IFIP TC6 / WG6.1 3rd International Working Conference on New Developments in Distributed Applications and Interoperable Systems*, Deventer, The Netherlands, September 2001.

41. K. Nahrstedt, D. Xu, D. Wichadakul, and B. Li. QoS-aware middleware for ubiquitous and heterogeneous environments. *IEEE Communications Magazine*, 39(11):140–148, November 2001.

42. J. Nord, K. Synnes, and P. Parnes. An architecture for location aware applications. In *Proc. of the 35th Hawaii International Conference on System Sciences*, Big Island, Hawaii, USA, January 2002.

43. P. Olofsson. *Probability, Statistics and Stochastic Processes*. John Wiley & Sons, Inc., 2005.

44. Openwave. *Openwave Location Manager*, 2006. http://www.openwave.com/.

45. B. Parkinson, J. Spilker, P. Axelrad, and P. Enge, editors. *Global Positioning System: Theory and Application, Volume II*, volume Progress in Austronautics and Aerounautics Series, V-164. American Institute of Astronautics and Aeronautics (AIAA), 1996.

46. Privacy Rights Clearinghouse/UCAN. *A Chronology of Data Breaches*, 2006. http://www.privacyrights.org/ar/ChronDataBreaches.htm.

47. A. Ranganathan, J. Al-Muhtadi, S. Chetan, R. H. Campbell, and M. D. Mickunas. Middlewhere: A middleware for location awareness in ubiquitous computing

applications. In *Proc. of the ACM/IFIP/USENIX 5th International Middleware Conference (Middleware 2004)*, Toronto, Ontario, Canada, October 2004.

48. N. Sastry, U. Shankar, and S. Wagner. Secure verification of location claims. In *Proc. of the ACM Workshop on Wireless Security (WiSe 2003)*, San Diego, CA, USA, September 2003.

49. K.E. Seamons, M. Winslett, T. Yu, L. Yu, and R. Jarvis. Protecting privacy during on-line trust negotiation. In *Proc. of the 2nd Workshop on Privacy Enhancing Technologies*, San Francisco, CA, April 2002.

50. G. Sun, J. Chen, W. Guo, and K. R. Liu. Signal processing techniques in network-aided positioning: A survey of state-of-the-art positioning designs. *IEEE Signal Processing Magazine*, 22(4):12–23, July 2005.

51. T.W. van der Horst, T. Sundelin, K.E. Seamons, and C.D. Knutson. Mobile trust negotiation: Authentication and authorization in dynamic mobile networks. In *Proc. of the 8th IFIP Conference on Communications and Multimedia Security*, Lake Windermere, England, September 2004.

52. U. Varshney. Location management for mobile commerce applications in wireless internet environment. *ACM Transactions on Internet Technology*, 3(3):236–255, August 2003.

53. G. Zhang and M. Parashar. Dynamic context-aware access control for grid applications. In *Proc. of the 4th International Workshop on Grid Computing (Grid 2003)*, Phoenix, Arizona, November 2003.