Spatiotemporal Access Control Enforcement under Uncertain Location Estimates

Heechang Shin and Vijayalakshmi Atluri

MSIS Department and CIMIC, Rutgers University, USA
{hshin,atluri}@cimic.rutgers.edu

Abstract. In a mobile environment, user's physical location plays an important role in determining access to resources. However, because current moving object databases do not keep the exact location of the moving objects, but rather maintain their approximate location for reasons of minimizing the updates, the access request evaluation cannot always guarantee the intended access control policy requirements. This may be risky to the system's security, especially for the highly sensitive resources. In this paper, we introduce an authorization model that takes the uncertainty of location measures into consideration for specifying and evaluating access control policies. An access request is granted only if the confidence level of the location predicate exceeds the predefined uncertainty threshold level specified in the policy. However, this access request evaluation is computationally expensive as it requires to evaluate a location predicate condition and may also require evaluating the entire moving object database. For reducing the cost of evaluation, in this paper, we compute lower and upper bounds (R_{min} and R_{max}) on the region that minimize the region to be evaluated thereby allowing unneeded moving objects to be discarded from evaluation. We show how R_{min} and R_{max} can be computed and maintained, and provide algorithms to process access requests.

1 Introduction

Unlike the traditional access control system, in a mobile environment, a person's physical location plays an important role in determining access rights. For example, access to certain important resources of an organization can be restricted to employees who are currently located within the office area. This concept of location-based access control (LBAC) system is not new. For example, in [6], when an access request is made from a moving object, the system checks whether the requester lies within the authorized region and only if this is the case, access is granted.

We can categorize access control rules based on the mobility of subject/resources as (i) *static subjects to access mobile resources* (SM), (ii) *mobile subjects to access static resources* (MS), and (iii) *mobile subjects to access mobile resources* (MM). First, SM is to restrict access to resources based on the spatiotemporal locations of resources. For example, the New York branch operations department of a truck company are allowed to track the locations of dispatched trucks that are currently located within New York City. Second, MS restricts access to static resources based on the spatiotemporal location of the subjects. For example, security managers are allowed to read or write the mobile network data only when they are currently located within the server farm room. Finally, in MM, access control decisions are made based on the spatiotemporal location of subjects and resources. For example, all supervisors who are currently located within the office building can locate their employees only when they are also currently in the same building.

The proposed LBAC systems [4, 6, 9, 18] assume that provided location measure of moving object is always accurate. For example, in GEO-RBAC [9], positions can be real or logical: the real position corresponds to the position on the Earth of the mobile user obtained from location-sensing technologies such as Global Positioning Systems (GPS) while the logical position is modeled as a polygon feature such as city, i.e., a real position acquired through GPS can be mapped to a corresponding road segment (logical position). A spatial role is activated based on the location (either logical or real) of the user. However, considering the fact that users are moving objects, most of the time, the provided location information is not precise because of continuous motion [21]. In fact, most of currently proposed moving object databases do not keep the exact location of the moving objects, rather maintain the approximate value of the location in order to minimize the updates. Therefore, a location measure stored in the moving object database should be modeled as a region instead. In general, we call this inherent error of a location measure as *location uncertainty*. However, if we consider the inherent uncertainty of location measures, the role activation in GEO-RBAC cannot guarantee the desired security. In other words, their underlying assumption that any logical position can be computed from real positions by using specific mapping functions are not true any longer because it is possible that several logical positions can be mapped from a single real position. This may incur huge risks to the security of the system especially for highly sensitive resources. Therefore, it is essential that all LBAC systems must incorporate the concept of uncertainty within the model.

To the best of our knowledge, the work of Ardagna et al. [3] is the only LBAC model where uncertainty is considered. They present a model for representing and evaluating a set of location predicates. Each access request can gain access to the specified resources only if the confidence level of the location predicate result exceeds the predefined uncertainty threshold level. Formally, given an access control rule's location predicate and a user o, we need to evaluate the probability p_o, the chance that o satisfies the given location predicate, to determine the satisfiability of the predicate (i.e, o is located within an authorized region R). Given an authorized region R and o's uncertainty region denoted as $o.ur$, p_o is computed as

$$p_o = \int_{o.ur \cap R} f_o(x)dx \tag{1}$$

where x is the location of o in d-dimensional data space D, f_o is the probability density function (PDF), and $o.ur \cap R$ is the intersection of $o.ur$ and R. In other words, p_o is the confidence level of the location predicate result. The user o gains access to the resources only if $p_o \geq p_c$ where p_c is the predetermined predicate threshold. However, their model has the following limitations: (i) the uncertainty thresholds for location predicates are globally fixed values, thus lacking the specification power for different situations: for example, the minimum threshold of location predicate for granting access is a globally fixed level, and therefore, it cannot differentiate between highly security-sensitive area and less sensitive area by assigning different confidence levels; and (ii) resources are assumed to be always static, and therefore, only MS type of security policies for a mobile environment can be evaluated.

To address the above limitations, we introduce an access control model that embeds uncertainty within the model. In this model, we allow varying threshold levels of location predicates; thus, it is possible to differentiate the highly security sensitive area and less sensitive area in an access control rule. Also, in addition to MS type of security policies, SM and MS type of policies are supported. Although Ardagna et al.'s model can be extended to support MM and SM type of queries by incorporating location predicates in specifying resources, the main challenge of supporting them in their model is the evaluation part, which is the focus of this paper: it is hard to evaluate the resources' satisfiability to an access request. For example, if location of employees in a given area are requested, considering the location uncertainty, the access control enforcement system cannot simply release the location of employees inside the region since it is still possible that some people who are believed to be located outside may, in fact, be inside, and vice versa, i.e., people believed to be inside may actually be outside. Under Ardagna et al.'s model, in order to guarantee the correctness of the query results, it may require to evaluate all the moving objects in the database since they only allow Boolean queries.

Our main objective in this paper is to reduce the cost of location predicate evaluation. There are two main challenges: (i) the computation of Equation (1) is computationally expensive. For example, under the normal distribution case, $o.ur \cap R$ is not symmetric with respect to the mean [20], and therefore, it is expensive to compute; (ii) the size of uncertainty region grows as time elapses because the actual location can deviate further than the measured one. This implies that location predicate evaluation cannot be computed in advance. In order to address the first issue, Yufei et al. [20] propose a Monte Carlo based approach. This method generates inputs randomly, performs a deterministic computation using the inputs and aggregates the results of the individual computations into the final result. But it is relatively accurate only if the sampled points are sufficiently large. This is because, the computation of probability is based on the sampling, the result of this approach is close to the actual value only if there are enough number of samples (i.e., at the order of 10^6 in their experimental study). Even worse when considering the second problem, in the moving object database, Equation (1) needs to be computed within the reasonable amount of time because the satisfying condition of location predicate changes as the position of o is constantly updated. Also, because the size of uncertainty region grows as time elapses after the last location update, it requires the continuous evaluation of the specified location predicates. Therefore, we need to reduce the cost of computation of p_o as much as possible.

Our proposed approach is to find the upper and lower bounds of the region to be evaluated, essentially identifying two regions: (i) the first, called R_{min}, is the region that guarantees the correctness of the location predicate evaluation if the location estimate is within this region, and (ii) the second, called R_{max}, is the region where any location measure outside of this region is guaranteed to have no probability to satisfy the given predicate. Once these regions are found, the cost of location predicate evaluation process is significantly reduced because it requires simple location containment test to evaluate the predicate correctness for most of location measures instead of expensive computation of Equation (1). More specifically, instead of computing p_o by using Equation (1) for all the location measures, we take the following steps:

1. Those objects which are located outside of R_{max} are filtered out from the candidate set for examining their satisfaction for the given predicate;
2. Among the objects located within R_{max}, we do not need to evaluate p_o of those objects located within R_{min} because it is guaranteed to have $p_o \geq p_c$;
3. p_o needs to be computed only for those objects located in $R_{max} - R_{min}$. In order to minimize the cost of p_o evaluation, our objective is to minimize the area size of $R_{max} - R_{min}$.

We discuss how to generate R_{max} and R_{min} while minimizing $R_{max} - R_{min}$. Specific probability distribution such as uniform distribution is considered to find R_{max} and R_{min}.

The rest of the paper is organized as follows. Section 2 introduces preliminaries of the paper. In Section 3, we present an access control model where the concept of uncertainty is embedded. Section 4 introduces our novel concept of R_{max} and R_{min} under specific probability distribution such as uniform distribution and presents our algorithm to handle imprecise access requests. Related work is presented in Section 5. Section 6 concludes the paper with some suggestions for future work.

2 Preliminaries

In this section, we describe the uncertainty model of moving objects and present an overview of our system architecture in a mobile environment.

2.1 Uncertainty of Moving Objects

According to [13], in the mobile network environment, no technology is available that ensures precisely the exact user locations. Thus, a position of a moving object, instead of a single location point, is rather specified with a range, called uncertainty region. The uncertainty is caused by the sampling error and the measurement error [17].

Sampling Error: It is unrealistic to obtain the current location of the moving objects continuously under the existing location sensing technologies and database technologies, and the position is collected at discrete instances of time such as every few seconds instead [17]. The solid line in Figure 1 represents the projected movement of a moving object in one dimensional space (x axis) and time (t axis). Linear interpolation is used to project positions between two consecutive location updates: the sampled positions become the end points of single line segments, and the entire polyline (i.e., solid line) represents the projected movement of the object. However, this approach brings the error due to the position estimation methods of moving objects within any single line segment except the end point. For example, in Figure 1, the dashed line shows the actual locations of the object between t_0 and t_5. After the location is updated, because

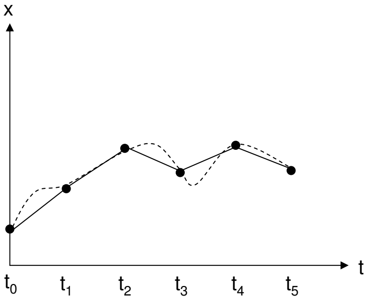

Fig. 1. Position History

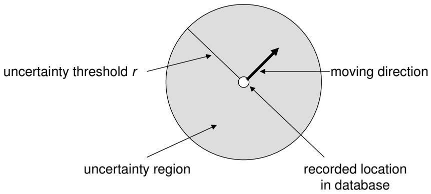

Fig. 2. Uncertain Object Example

the position of the moving object is unknown until the next location update, the actual location can be anywhere within the so called uncertain region.

Measurement Error: Location sensing techniques determine the accuracy and the quality of the location measurements. Pfoser et al. [17] propose that given a location measure $x = (\mu_{x_1}, \mu_{x_2})$ in 2-dimensional space, the error in a positional GPS measurement can be described by the bivariate normal distribution where the mean (or variance) of the distribution is the measured location of the coordinate system (or $\sigma = \sigma_{x_1} = \sigma_{x_2}$), and the covariance is 0. More specifically, a positional GPS measurement is described by

$$f_o(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x_1-\mu_{x_1})^2+(x_2-\mu_{x_2})^2}{2\sigma^2}} \tag{2}$$

This implies that the distribution in the x_1-x_2 plane is circular, and within the range of $\pm\sigma$ of the mean, 39.35% of the probability is concentrated. Depending on the location sensing techniques, the value of σ can be determined. Given $\sigma_1 < \sigma_2$ with the same location measure, it is obvious that the location measure involved with σ_2 would involve more measurement error. Liu et al. [15] use two performance metrics for measurement error:

- *Accuracy* is the degree of closeness between the estimated location and the true location.
- *Precision* is the distribution error between the estimated location and the true location. In other words, it measures how consistently the system measures the given location.

For example, HORUS [26, 24] has the accuracy of 2m and the precision of 90% within 2.1m [15], and the maximum measurement error level can be specified as 2m + 2.1m = 4.1m with 90% certainty.

Due to the above errors, the system does not have users' precise positions, and the user can be anywhere in an uncertainty region.

Definition 1 (Moving Object Uncertainty). *Given a set of moving objects O, uncertainty of a moving object $o \in O$ in the d-dimensional data space D is conceptually* *described by (i) a d-dimensional uncertainty region, denoted by $o.ur$ and (ii) a PDF f, denoted by $f_o(x)$ where $x \in D$ is the d dimensional location.*

Notice that $f(x) \geq 0$ for any point $x \in D$ and $\int_{o.ur} f_o(x)dx = 1$. Also, $f_o(x) = 0$ if o is located outside of $o.ur$. We use $loc(o)$ to denote the last update location of o in the database. Uncertainty region is often represented with a circle with uncertainty threshold r, which is determined by

$$r = m_{error} + v_{max} \cdot |t_c - t_u| \tag{3}$$

where m_{error} is the measurement error level of location sensing technologies, v_{max} is the maximum speed, t_c is the current time, and t_u is the last update time. In other words, the circle with radius r is the region that a user can be possibly located after the last location update. Figure 2 illustrates such an example. Observe that m_{error} refers to the measurement error and $v_{max} \cdot |t_c - t_u|$ refers to the sampling error. Since the PDF of an uncertain object is often unavailable explicitly, a set of samples are drawn or collected in the hope of approximating the PDF [16]. However, in some applications, it would make sense to use the specific PDF f for specifying the uncertainty region. For example, Wolfson et al. [19] propose that the object location follows the Gaussian distribution over the uncertainty region. Also, uniform distribution is used in many application scenarios [17, 21] to represent an uncertainty region.

2.2 System Architecture Overview

We assume the system in a mobile environment comprises of the following components:

Location Server (LS): The main objective of LS is to enforce access control policies in order to protect the important resources. Location information can be maintained by either LS itself or another trusted third party such as a mobile service provider. The current location of moving objects are stored and updated accordingly in order to provide most up-to-date location information to a service requester. If LS maintains the location information, users' mobile devices directly provide such information via wireless communication periodically, the installed sensors can approximate it. For example, the Active Badge [1] detects the location of each user in a building. Each individual carries a device called, badge, which is associated with the identifier of the user. A building is equipped with sensors detecting positions of badges. A person's location is determined by using an active badge which emits a signal every 15 seconds.

Requester: A requester subscribes to a service in order to gain access to the resources. In a mobile environment, there are two types of resources that a requester can gain access to: static resources (e.g. repository room or printer) and mobile resources (location of vehicles). For example, consider a work environment where all the documents can only be accessed by employees only while they are physically located in the office. Similarly, requester can be either static or mobile. For example, when a mobile requester submits an access request to the documents in the repository, LS checks the physical location of the requester, and only if the subject is within premises of the office, he is given an access. However, in some cases, the location of requestors is always fixed. For example, an emergency message (e.g., reverse 911) need to be delivered to students' mobile devices by the university police office, only while the students' are on campus.

Under our framework, LS is responsible for enforcing security policies. More specifically, when a user (mobile or static) submits an access request, the LS searches relevant security policies that are applicable to the submitted access request. If a location predicate is included in a relevant policy, location information is either provided by the LS (if the location information maintained by the LS) or retrieved from the trusted location service provider. The provided location measures should compute the inherent uncertainty of provided location measures (discussed in Section 2.1). After evaluating the policies, the LS returns answers to the requester.

3 Location-Based Access Control Model under Uncertain Location Estimates

In this section, we introduce a location-based access control model suitable for moving object data with inherent uncertainty by extending the GSAM [4, 5]. An access control rule, in general, is specified on the basis of three parameters, $\langle s, o, p\rangle$ which states that s is authorized to exercise privilege p on resource o[1]. However, this basic access control rule lacks specification power to include moving object data since an access control rule should be capable of specifications based on spatiotemporal attributes of both subjects and resources that are functions of time. In the following, we extend the basic authorization to accommodate this.

Definition 2 (Access Control Rule). *An access control rule is a triple of the form ⟨se, re, pr⟩ where se is a subject expression that denotes a set of authorized subjects, re is a resource expression that denotes a set of authorized resources, and pr is a set of privilege modes that denotes the set of allowed operations.*

In this paper, we use $\mathcal{P}$ to denote the set of access control rules stored in the LS. Given an access control rule $\alpha \in \mathcal{P}$, *se(α), re(α) and pr(α)* denote the set of subjects satisfying subject expression, the set of resources satisfying resource expression, and the set of privileges, respectively, of α. Also, $\alpha(R_r)$ and $\alpha(R_s)$ denote the authorized region specified in se(α) and re(α), respectively. In the following, we discuss these concepts in detail.

Definition 3 (Subject Expression). *A subject expression is a triple of the form ⟨R, sc, ue⟩ where R is the role to which the subject belongs, sc is the location predicate, called* scene*, which can be associated with a set of geospatial and temporal extents, and ue is an uncertainty expression associated with a* scene.

Similar to subject expression, a resource expression includes (i) an object type t which evaluates the membership of the object in categories, or values of properties on metadata, and (ii) location predicate with uncertainty expression.

[1] 'Object' is more general term to specify o, but in order not to confuse with moving objects, we specify o as resources throughout the paper.

Definition 4 (Resource Expression). *A resource expression is a triple of the form ⟨t, sc, ue⟩ where t the object type to which the resource belongs, sc is a location predicate, called scene, which can be associated with a set of geospatial and temporal extents, and ue is an uncertainty expression associated with a scene.*

In this paper, we assume the formalism developed in [5] to specify *R* and *sc* in a subject expression. Due to space limitations, we do not review the details. In short, *R* refers to a role in RBAC with roles organized as a hierarchy, and *sc* is a conceptual event or region that can be mapped to a set of bounding boxes represented with $\langle label, lt, lg, h, w, [t_b, t_e]\rangle$ where *label* is a descriptive scene name, $\langle lt, lg, h, w\rangle$ denotes latitude, longitude, height and width of a bounding box covering a geographic area of the *scene* during temporal period between t_b and t_e[2]. An object type o in an object expression can be organized into a hierarchy similar to a role hierarchy. In [5], only geospatial objects are considered, but it can be extended to support other types of objects as well. Both subject expression and object expression can also contain more complex static predicates or expressions other than roles if necessary.

An uncertainty expression *ue* is a logical expression denoting uncertainty level of the corresponding *scene* in both se and re. As we have discussed in Section 2.1, any location measure stored in the database includes inherent uncertainty.

Definition 5 (Uncertainty Expression). *Given $\alpha \in \mathcal{P}$ and a finite set of scenes $S = \{sc_1, sc_2, \cdots, sc_m\}$ used in se(α) or re(α), an uncertainty expression, denoted as* ue*(α), is defined as follows:*

- *If sc_i is a scene, op $\in \{=, \neq, <, >, \leq, \geq\}$, and p_c is a real number in the range from 0 to 1, sc_i op p_c is a ue.*
- *If ue_1 and ue_2 are two uncertainty expressions, $ue_1 \wedge ue_2$, $ue_1 \vee ue_2$, $\neg ue_1$, and (ue_1) are ue.*

Although we allow any logical operator (i.e, $=, \neq, <, >, \leq, \geq$) for specifying *ue*, we particularly focus on $\geq$ operator in this paper since it plays an important role to prune out moving objects (either subjects or resources) that do not satisfy the uncertainty threshold specification (i.e., p_c). Evaluation of *scene* (location predicate) results in the following form [*result_set, timeout*] stating that each element in the *result_set* includes a moving object and its corresponding confidence level, and every element in the *result_set* exceeds the specified uncertainty threshold level in the corresponding uncertainty expression. More specifically, the confidence value of each object o, denoted as p_o, is compared with the predetermined value of threshold in *ue*, denoted as p_c, and those objects whose confidence level is greater than the threshold are included in the *result_set*. Although [3] requires two thresholds for accepting or rejecting the evaluation, the request is granted only if the confidence level is above the upper threshold.

[2] Actually, *sc* corresponds to the inarea() location predicate in [2]. In this paper, we mainly focus on this type of predicate evaluation because other location predicates introduced in [2] is a special case of the proposed approach. For example, in case of velocity, it is the special case of the proposed approach with one dimensional space, and thus, the proposed approach is general enough to evaluate other location predicates.

Therefore, without loss of generality, we only require one threshold for evaluating location predicates. Also, the timeout represents the time validity of the result. This timeout takes into account that location values may change rapidly, even during policy evaluation. After it is expired, *scene* must be reevaluated to guarantee the correctness of the evaluation since the location measures are constantly updated.

- $\alpha_1 = \langle\{\text{operations\_dept}(x)\}, \{\text{truck}(y), \text{New\_York\_City}(y), \text{New\_York\_City} \geq 0.7\}, \{\text{track}\}\rangle$: This rule states that the operations department can track the locations of dispatched trucks that are currently located within New York City with greater than 70 % confidence.
- $\alpha_2 = \langle\{\text{Security\_manager}(x), \text{server\_farm\_room}(x), \text{server\_farm\_room} = 1.0\}, \{\text{mobile\_network}(y)\}, \{\text{read} \wedge \text{write }\}\rangle$: This rules states that all security mangers who are currently located within the server farm room with confidence level of 100% can read or write the mobile network data.
- $\alpha_3 = \langle\{\text{Supervisor}(x), \text{office\_building}(x), \text{office\_building} \geq 0.8\}, \{\text{employee}(y), \text{office\_building}(y), \text{office\_building} \geq 0.9\}, \{\text{locate}\}\ \rangle$: This rules states that all supervisors who are currently located within the office building with confidence level greater than 80 % can locate their employees who are also currently located within the building with greater than 90 % confidence level.

The access control rules α_1, α_2, α_3 refer to SM, MS, and MM type respectively. Each access control rule has its own uncertainty level specified for location predicates. We consider that the network configuration in a server farm room in α_2 is the most highly sensitive to security because configuration must be performed according to the highest security standards. Therefore, 100% of location confidence should be guaranteed to do such a job. Accessing the locations of employees in the office building is considered 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 [2]. Finally, we consider tracking dispatched trucks for operational purposes as the lowest critical to security.

4 Security Policy Evaluation of Uncertain Location Samples

In this section, we present our proposed approach that evaluates access control policies efficiently. Then, our solutions considering certain reasonable assumptions are described.

4.1 Proposed Approach

Our approach is to find two regions: (i) the first region, called R_{min}, is the region that guarantees the correctness of the location predicate evaluation if the location estimate is located within this region, and (ii) the second region, called R_{max}, is the region where any location measure located outside of this region is guaranteed to have no probability to satisfy the given predicate. Formally, given an authorized region $R = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_d, b_d] \subseteq D$ where $a_i < b_i$ for $i = 1, 2, \cdots, d$, we want

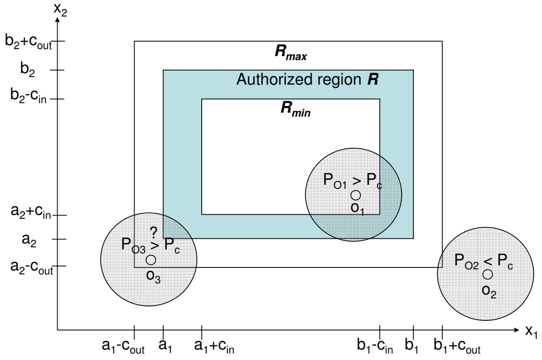

Fig. 3. Use of R_{min} and R_{max}

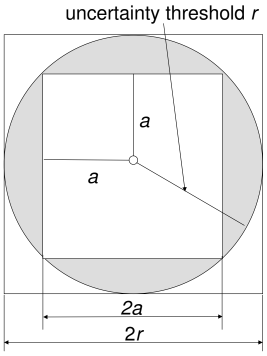

Fig. 4. Approximation of Uncertainty Region

to find $c_{in}, c_{out} \geq 0$ such that $R_{min} := [a_1 + c_{in}, b_1 - c_{in}] \times [a_2 + c_{in}, b_2 - c_{in}] \times \cdots \times [a_d + c_{in}, b_d - c_{in}]$ $(a_i + c_{in} < b_i - c_{in}$ for $i = 1, 2, \cdots, d)$ and $R_{max} := [a_1 - c_{out}, b_1 + c_{out}] \times [a_2 - c_{out}, b_2 + c_{out}] \times \cdots \times [a_d - c_{out}, b_d + c_{out}]$ $(a_i - c_{out} < b_i + c_{out}$ for $i = 1, 2, \cdots, d)$ where $\forall o \in O$, $\min(p_o) \geq p_c$ if loc(o) is contained in R_{min} and $\max(p_o) \leq p_c$ if loc(o) is contained in $D - R_{max}$.

Figure 3 illustrates R_{min} and R_{max}. In case of R_{min}, the original authorized region R is reduced to R_{min} by c_{in} in every dimension so that if any location measure loc(o) that is stored in the last update is contained within R_{min}, we can guarantee that $p_o \geq p_c$. Therefore, in case of o_1, it is guaranteed to have $p_{o_1} \geq p_c$ because loc(o_1) is located within R_{min}. Similarly, any location measure loc(o) outside of R_{max} is guaranteed to satisfy $p_o < p_c$. For example, o_2 is located outside of R_{max}, i.e., loc(o_2)is contained in $D - R_{max}$, and thus, $p_{o_2} < p_c$ holds. However, we do not know how the result of location predicate evaluation for any location measure located in $R_{max} - R_{min}$. Therefore, in this case, we have to manually compute p_o for any $o \in O$ located within this region, i.e., we should evaluate p_{o_3} in order to see if $p_{o_3} \geq p_c$ holds. This example illustrates that it is important to have $R_{max} - R_{min}$ as small as possible. In other words, our objective is to compute the minimized value of c_{in} and c_{out} and therefore, the number of computations for Equation (1) is minimized as well.

Obviously, the main benefit of our proposed approach is that the cost of location predicate evaluation process is significantly reduced once R_{min} and R_{max} are computed because it requires simple location containment test to evaluate the predicate's correctness for most of location measures instead of expensive computation of Equation (1). Throughout the paper, we restrict our discussion on 2-dimensional space for its easiness to illustration. However, it is simple to extend to a higher dimensional space.

4.2 Uniform Distribution Case

Here we discuss how to compute the value of c_{in} and c_{out} to find corresponding R_{min} and R_{max} when uncertainty region is approximated with square under the assumption of uniform distribution.

Approximation of Uncertainty Region: Suppose a security policy is evaluated over a moving object $o \in O$ in 2-dimensional data space D. For example, o is one of the resources (or subjects) in SM (or MS) type of policies while o can be both resources and subjects in MM. Also, an authorized region R is specified in either se or re or both. In case of a circular shape of $o.ur$ centered at (μ_{x_1}, μ_{x_2}) with radius r ($\mu_{x_1}, \mu_{x_2} \geq 0$), its PDF $f_o(x_1, y_1)$ ($x_1, y_1 \geq 0$) is defined as

$$f(x_1, x_2) = \begin{cases} g(x_1, x_2) & \text{if}(x_1 - \mu_{x_1})^2 + (x_2 - \mu_{x_2})^2 \leq r^2 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $g(x_1, x_2)$ is a probability distribution such as uniform or bivariate normal distribution. Then, p_o is defined as

$$p_o = \int_{\max(a_1,\mu_{x_1}-r)}^{\min(b_1,\mu_{x_1}+r)} \int_{\max(a_2,\mu_{x_2}-\sqrt{r^2-(x_1-\mu_{x_1})^2})}^{min(b_2,\mu_{x_2}+\sqrt{r^2-(x_1-\mu_{x_1})^2})} f(x_1, x_2)\, dx_2\, dx_1 \tag{5}$$

However, it turns out that even with the uniform distribution, the evaluation of Equation (5) is very expensive. We can use square instead of circle for representing an uncertainty region for computing R_{min} and R_{max}. Figure 4 illustrates that we use two squares to approximate the circular shape of uncertainty region. For example, in case of R_{min}, we can use a rectangle whose circumcircle passes through all the vertices of it: the inner square with edge's size= $2a$ where $a = \frac{r}{\sqrt{2}}$ because the radius of the circumcircle is the same as the radius of the polygon as shown in the figure. This is because we want to have the minimum value of p_o for objects located within R_{min} satisfy $p_o \geq p_c$. Then, the size of area($R \cap o.ur$) is getting smaller, implying that p_o is getting smaller compared to the original value of p_o. Given an uncertainty region $o.ur$ with circle shape where the center is (μ_{x_1}, μ_{x_2}) and the radius of r, the corresponding rectangle becomes $[\mu_{x_1} - \frac{r}{\sqrt{2}}, \mu_{x_1} + \frac{r}{\sqrt{2}}] \times [\mu_{x_2} - \frac{r}{\sqrt{2}}, \mu_{x_2} + \frac{r}{\sqrt{2}}]$.

In case of R_{max}, we want to have $\forall o \in O\ p_o \leq p_c$ satisfied. Thus, we want to find a rectangle whose incircle is $o.ur$, illustrated with an outer rectangle in Figure 4. Then, the area($R \cap o.ur$) is getting larger, implying that p_o is also getting larger. Because the incircle's radius is the apothem of the rectangle, the corresponding rectangle becomes $[\mu_{x_1} - r, \mu_{x_1} + r] \times [\mu_{x_2} - r, \mu_{x_2} + r]$ when the uncertain region's circle with the center (μ_{x_1}, μ_{x_2}) and the radius of r.

Now, we are ready to discuss how to compute the value of c_{in} and c_{out} to find corresponding R_{min} and R_{max} under the uniform distribution assumption. We represent the uncertainty region as square where each side's width is $2r$ without loss of generality for simple representation. Under the uniform distribution assumption of uncertainty region which is represented with rectangular shape,

$$p_o = \frac{area(o.ur \cap R)}{area(o.ur)} \tag{6}$$

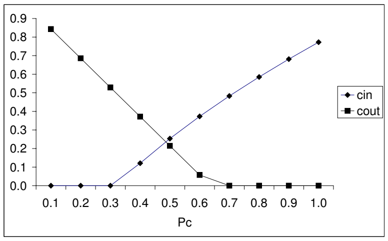

Fig. 5. Effect of p_c when $r = 1$

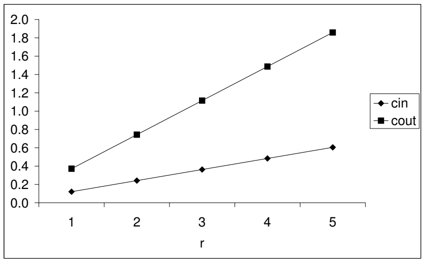

Fig. 6. Effect of r when $p_c = 0.4$

where *area*() returns the area of the given region.

Finding Minimal c_{in}: Let w and h be the width and height of $o.ur \cap R$, respectively, implying $area(o.ur \cap R)= w \cdot h$. In order to find the minimized value of c_{in}, consider the case where p_o is minimized but still satisfies $p_o \geq p_c$, i.e., for all o_i located within R_{min}, $\min_i(p_{o_i}) \geq p_c$ is satisfied. In this case, we can set $m = \min(w, h)$ without loss of generality. Then, from Equation (6), the inequality $\frac{m^2}{area(o.ur)} \geq p_c$ holds, implying $\frac{(r+c_{in})^2}{\pi r^2} \geq p_c$ since $m = r + c_{in}$ when p_o is minimized. Because we want to find the minimum value of c_{in}, we can set $c_{in} = \max(r(\sqrt{\pi p_c} - 1), 0)$. Therefore, R_{min} is computed by shrinking R by c_{in} in each dimension. One thing to notice is that we cannot fix the value of c_{in} in advance because the size of uncertainty is dependent on the elapsed time after the last update as illustrated in Section 2.1.

Finding Minimal c_{out}: In order to find the minimal value of c_{out}, consider the case where p_o is maximized while still satisfying $p_o \leq p_c$. Let w and h be the width and height of $o.ur \cap R$. Observe that p_o is maximized when $w = 2r$ and $h = r - c_{out}$ or vice versa.[3] In this case, $area(o.ur \cap R) = 2r(r - c_{out})$. Then, the inequality $2r(r - c_{out}) \leq p_c \pi r^2$ holds from Equation (6), which implies $c_{out} \geq (1 - \frac{1}{2}\pi p_c)r$. Because we want to find the minimum value of c_{out}, we can set $c_{out} = \max((1 - \frac{1}{2}\pi p_c)r, 0)$.

Example 1. Given $R = [10, 20] \times [10, 20]$, $p_c = 0.4$, and $r = 1$, c_{in} and c_{out} become $c_{in} = 0.1270$ and $c_{out} = 0.3717$. Thus, R_{min} is $[10.1210, 19.8790] \times [10.1210, 19.8790]$, and $R_{max} = [9.6283, 20.3717] \times [9.6283, 20.3717]$.

Figure 5 and Figure 6 show the experimental results of c_{in} and c_{out} for various values of p_c and r respectively. As Figure 5 illustrates, the value of c_{in} is increased while the value of c_{out} is decreased with respect to the increasing value of p_c when r is fixed as 1. In case of c_{in}, this is expected because in order to have R_{min} guarantee the correctness of the given location predicate evaluation, the size of R_{min} should be

[3] This is because of an implicit assumption of the condition $c_{out} \leq r$ since we want to have area(R_{max}) $\leq$ area(R).

smaller (or c_{in} is increased) as p_c is increased. However, the size of R_{max} gets smaller (or c_{out} gets smaller) as the value of p_c is increased, which is also expected because higher threshold value implies that smaller number of objects satisfy this threshold. Figure 6 illustrates that both c_{in} and c_{out} is increased with respect to the increasing value of r when $p_c = 0.4$. This is because both c_{in} and c_{out} has the positive relationship with r in the formulae.

4.3 Evaluation of Imprecise Access Requests

The imprecision of location measures implies that given $\alpha \in \mathcal{P}$, either $se(\alpha)$ if location predicate is included, or $re(\alpha)$, if location predicate is included, cannot be evaluated deterministically. Instead, it is natural to assign a probability value to the requester's answer to access request results. This approach is similar to uncertain databases [22] where each object in the query results is associated with the probability value such as (o_i, p_i) where o_i is the object and p_i is the quantification of probability that o_i satisfies the query.

Definition 6 (Imprecise Access Request). *An imprecise access request (IAR) is the form of $\langle user\_id, re,\ action, p_q\rangle$ where user_id is the identifier of the user who submits the request, re is a resource expression, action is the requested action, and p_q is the probability threshold that the probability quantification (p_i) of each resource that evaluates re be true must satisfy (i.e., $p_i \geq p_q$ must holds).*

Given an IAR q, $user\_id(q)$, $re(q)$, $action(q)$, $p_q(q)$ will denote the user, the action, the set of resources evaluated by the resource expression, and p_q of q. The result of IAR is a set of resources that are allowed to gain access to and their quantification probability p_i is greater than or equal to p_q. Given an IAR, LS evaluates $\mathcal{P}$ to find all the relevant rules $\mathcal{P}' \subseteq \mathcal{P}$ that are applicable to the requester. Then, we need to find if $\exists \alpha \in \mathcal{P}'$, the objects specified by re(α) contains each u specified by re(q). Algorithm 1 describes the detailed IAR processing by utilize of the proposed R_{min} and R_{max}.

5 Related Work

Incorporating location information for access control is not a new concept. Atluri and Chun [4] propose an access control model suitable to geo-spatial data. Similarly, Bertino et al. [9] extends the RBAC model to support location-based conditions, called GEO-RBAC, which can deal with mobile data. However, these models do not consider uncertainty within the model, and thus, the access control decision does not guarantee the correctness of the evaluation. There are also several access control models that support protecting people location information in the context of ubiquitous or pervasive computing environment [12, 14]. However, these approaches are different from our approach because they focus on preventing location information from leaking to unauthorized entities by introducing the concept of trust. Actually, Ardagna et al. [3] address the representation and evaluation of location-based access control systems with uncertainty considered, but it does not discuss efficient evaluation of access control requests.

Algorithm 1 IAR Processing

```
Input: a set of authorizations $\mathcal{P}$ and IAR $q$
Output: a set of authorized objects $O' \subseteq O$ where for each $o_i \in O$, $p_i \geq p_q(q)$.
for each $\alpha \in \mathcal{P}$ do
    if $q$ is MM or MS then
        retrieve the uncertainty region $o.ur$ of $user\_id(q)$ from the LS
        compute $R_{max}$ and $R_{min}$ from $\alpha(R)$
        if loc($o$) is contained in$R_{min}$ then
            $O_c \leftarrow O_c \cup$ re($\alpha$)
        else if  loc($o$) is contained in $D - R_{max}$ then
            do nothing
        else if $p_o \geq p_c$ then
            $O_c \leftarrow O_c \cup$ re($\alpha$).
        end if
    else
        if se($\alpha$) includes $user\_id(q)$ then
            $O_c \leftarrow O_c \cup$ re($\alpha$)
        end if
    end if
end for
if $q$ is SM or MM then
    Compute $R_{max}$ and $R_{min}$ from $\alpha(R_r)$
    Range query of resources located within $R_{min}$, denoted as $O_{c_1}$
    Range query of resources location within $R_{max} - R_{min}$, denoted as $O_{c_2}$.
    $O' \leftarrow$ result of set intersection operation of $O_{c_1}$ and $O_c$
    $O'' \leftarrow$ result of set intersection operation of $O_{c_2}$ and $O_c$
    for each $o \in O''$, remove it from $O''$ if $p_o < p_c$
    return $O' \cup O''$
else
    $O' \leftarrow$ result of set intersection operation of $O_c$ and re($q$)
    return $O'$
end if
```

Regarding the uncertainty, Wolfson et al. [23] introduce a cost based approach to determine the size of the uncertainty area. A formal quantitative approach to the aspect of uncertainty in modeling moving objects is presented in [17]. However, the authors limit the uncertainty to the past of the moving objects and the error may become very large as time approaches now. Trajcevski et al. [21] introduced a set of spatiotemporal range queries that apply the uncertainty in traditional range queries. Cheng et al. [10] are the first to formulate the uncertain data retrieval, and contrary to the case of traditional data, uncertain data retrieval involves probabilistic quality with the query results. The work in [11] develops the notion of *x-bounds*, and based on this concept, index-based access methods, called the probability threshold index for one-dimensional uncertain objects. Tao et al. develop a multi-dimensional access method, called the U-Tree [20] which extends [11] to multi-dimensional space.

6 Conclusions and Future Work

In this paper, we presented a solution to enforce access control policies suitable for a mobile environment with considering location uncertainty. Our proposed solution includes the uncertainty-embedded authorization model, efficient enforcement algorithms, and handling user requests. Our proposed R_{min} and R_{max} can effectively filter out most of the objects from the candidate answer so that evaluation of expensive computation is greatly reduced.

Several open issues still remain. A particularly interesting issue is how to create an index structure for authorizations in order to achieve efficient search process of relevant authorizations given an access request. Most of the currently available authorization enforcement techniques search all the authorization base to find relevant authorizations, which is not efficient obviously. Although there are some work for this direction such as [6, 7, 8, 25], no work has considered uncertainty issue. We would like to develop enforcement algorithms based on the proposed index structure by utilizing the concepts of R_{min} and R_{max}.

References

[1] Active badge next generation applications. Website. `http://www.cs.agh.edu.pl/ABng/applications.html`.

[2] C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. Location Privacy Protection Through Obfuscation-based Techniques. *IFIP TC11/WG 11.3 21st Annual Conference on Data and Applications Security*, 2007.

[3] C.A. Ardagna, M. Cremonini, E. Damiani, S.D.C. di Vimercati, and P. Samarati. Supporting location-based conditions in access control policies. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 212–222. ACM New York, NY, USA, 2006.

[4] V. Atluri and S.A. Chun. An Authorization Model for Geospatial Data. *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING*, pages 238–254, 2004.

[5] V. Atluri and S.A. Chun. A geotemporal role-based authorisation system. *International Journal of Information and Computer Security*, 1(1):143–168, 2007.

[6] V. Atluri and Q. Guo. Unified Index for Mobile Object Data and Authorizations. *LECTURE NOTES IN COMPUTER SCIENCE*, 3679:80, 2005.

[7] V. Atluri and H. Shin. Efficient Security Policy Enforcement in a Location Based Service Environment. *LECTURE NOTES IN COMPUTER SCIENCE*, 4602:61, 2007.

[8] V. Atluri, H. Shin, and J. Vaidya. Efficient security policy enforcement for the mobile environment. *Journal of Computer Security*, 16(4):439–475, 2008.

[9] E. Bertino, B. Catania, M.L. Damiani, and P. Perlasca. GEO-RBAC: a spatially aware RBAC. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 29–37. ACM New York, NY, USA, 2005.

[10] R. Cheng, D.V. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 551–562. ACM New York, NY, USA, 2003.

[11] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J.S. Vitter. Efficient indexing methods for probabilistic threshold queries over uncertain data. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 876–887. VLDB Endowment, 2004.

[12] U. Hengartner and P. Steenkiste. Access control to people location information. *ACM Transactions on Information and System Security (TISSEC)*, 8(4):424–456, 2005.

[13] S. Horsmanheimo, H. Jormakka, and J. Lähteenmäki. Location-Aided Planning in Mobile NetworkTrial Results. *Wireless Personal Communications*, 30(2):207–216, 2004.

[14] L. Kagal, T. Finin, and A. Joshi. Trust-Based Security in Pervasive Computing Environments. 2001.

[15] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1067–1080, 2007.

[16] J. Pei, M. Hua, Y. Tao, and X. Lin. Query answering techniques on uncertain and probabilistic data: tutorial summary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1357–1364. ACM New York, NY, USA, 2008.

[17] D. Pfoser and C.S. Jensen. Capturing the Uncertainty of Moving-Object Representations. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 111–131, 1999.

[18] I. Ray and M. Toahchoodee. A Spatio-temporal Role-Based Access Control Model. *LECTURE NOTES IN COMPUTER SCIENCE*, 4602:211, 2007.

[19] PA Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Querying the uncertain position of moving objects. *Temporal Databases: Research and Practice*, 1399, 1998.

[20] Y. Tao, R. Cheng, X. Xiao, W.K. Ngai, B. Kao, and S. Prabhakar. Indexing multi-dimensional uncertain data with arbitrary probability density functions. In *Proceedings of the 31st international conference on Very large data bases*, pages 922–933. VLDB Endowment, 2005.

[21] G. Trajcevski, O. Wolfson, F. Zhang, and S. Chamberlain. The Geometry of Uncertainty in Moving Objects Databases. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 233–250, 2002.

[22] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. CIDR, 2005.

[23] O. Wolfson and H. Yin. Accuracy and Resource Consumption in Tracking and Location Prediction. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 325–343, 2003.

[24] M. Youssef and A. Agrawala. Handling samples correlation in the horus system. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, 2004.

[25] M. Youssef, V. Atluri, and N.R. Adam. Preserving mobile customer privacy: an access control system for moving objects and customer profiles. In *Proceedings of the 6th international conference on Mobile data management*, pages 67–76. ACM New York, NY, USA, 2005.

[26] M.A. Youssef, A. Agrawala, and A.U. Shankar. WLAN location determination via clustering and probability distributions. In *IEEE PerCom 2003*, pages 23–26, 2003.