Seattle, Washington, USA
November 3, 2009

SPRINGL 2009

Proceedings of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS

Sponsored by:

ACM SIGSPATIAL

The Association for Computing Machinery, Inc.
1515 Broadway
New York, New York 10036

ACM ISBN: 978-1-60558-853-7/09/11

Additional copies may be ordered prepaid from:

	Phone: 1-800-342-6626
ACM Order Department	(U.S.A. and Canada)
P.O. BOX 11405	+1-212-626-0500
Church Street Station	(All other countries)
New York, NY 10286-1405	Fax: +1-212-944-1318
	E-mail: acmhelp@acm.org

Printed in the U.S.A.

Foreword

These proceedings contain the papers selected for presentation at the second edition of ACM *Workshop on Security and Privacy in GIS and LBS (SPRINGL 2009)* which is being held in conjunction with the ACM SIGSPATIAL GIS conference.

The aim of the workshop is to lay the foundation and agenda for research and development in the area of geospatial data security and privacy. Today's world is witnessing a dramatic increase and dissemination of geospatial data in several application contexts including homeland security, environmental crises, and natural and industrial disasters. Geospatial infrastructures are being leveraged by companies to provide a large variety of location-based services (LBS) able to tailor services to users. However, despite the increase of publicly accessible geospatial information only little attention is being paid on how to secure geospatial information systems (GIS) and LBS. Privacy is also of increasing concern given the sensitivity of personally-identifiable location information. This is despite major advancements that have been made in secure computing infrastructures and the secure and privacy-preserving management of traditional (relational) data in particular. Given these pressing needs for securing GIS and LBS as well as assuring privacy, it is compelling to investigate security and privacy aspects as they relate to the management of geospatial data and the development of both emerging LBS and mission-critical geographic applications.

This year's program is organized in three sessions: security policies and context-based security for GIS; location anonymization; and privacy and location trustworthiness. The program also features two outstanding invited speakers: Michael Gertz, from the University of Heidelberg, Germany, and Walid Aref from Purdue University, USA. The program is complemented by a panel focusing on inter-disciplinary research on privacy.

We hope that these proceedings will serve as a valuable reference for researchers and practitioners and facilitate further research and development in geospatial data security and privacy.

Putting together *SPRINGL 2009* has been a team effort. First of all, we would like to thank the authors for providing the content of the program. Additionally, we would like to express our gratitude to the program committee for a careful review process. Many thanks are due to Chenyun Dai of Purdue University for his excellent work on organizing and managing the workshop web site and to Ashish Kamra for assembling the workshop proceedings. We also gratefully acknowledge the organization committee of ACM GIS 2009 for the help with the workshop planning. Finally, we would like to express our appreciation to the workshop sponsors CERIAS, IBM, ACM SIGSPATIAL, the University of Milan and the European Research Project MODAP. Last but not least, we would like to thank Elisa Bertino, the general chair of *SPRINGL 2009*, for her guidance and support throughout the review process and program preparation.

We hope that you will find this program interesting and thought-provoking, and that the workshop will provide you with an opportunity to share ideas with other researchers and practitioners from institutions around the world. Enjoy!!

Maria Luisa Damiani
SPRINGL 2009 Co-Program Chair
Università degli Studi di Milano, Italy

Yucel Saygin
SPRINGL 2009 Co-Program Chair
Sabanci University, Istanbul, Turkey

Table of Contents

SPRINGL 2009 Organization vi

Keynote Presentation - I

The Role of Security in Scientific Data Management 1
Michael Gertz (University of Heidelberg, Germany)

Session 1: Security Policies and Context-based Security for GIS

Security Policies for the Visualization of Geo Data 2
Patrick Capolsini and Alban Gabillon (University of Polynésie Française)

A Generalized Context-based Access Control Model for Pervasive Environments. 12
José Bringel (University of Grenoble, France) and Hervé Martin (University of Grenoble, France)

Towards Location-Based Access Control in Healthcare Emergency Response.....22
Carmen Ruiz Vicente (Aalborg University, DK), Michael Kirkpatrick (Purdue University, USA), Gabriel Ghinita (Purdue University, USA), Elisa Bertino (Purdue University, USA) and Christian Jensen (Aalborg University, DK)

Session 2: Location Anonymization

Movement Data Anonymity Through Generalization.....27
Gennady Andrienko (Fraunhofer Institute,Germany), Natalia Andrienko (Fraunhofer Institute, Germany), Fosca Giannotti (ISTI-CNR Pisa, Italy), Anna Monreale (University of Pisa, Italy) and Dino Pedreschi (University of Pisa, Italy)

Protecting Location Privacy Against Spatial Inferences: The PROBE Approach.....32
Maria Luisa Damiani (University of Milan, Italy), Elisa Bertino (Purdue University, USA) and Claudio Silvestri (University of Venice, Italy)

Keynote Presentation - II

Location-aware Privacy and More: A Systems Approach using Context-aware Database Management Systems. 42
Walid Aref (Purdue University, USA), Hicham G. Elmongui (Purdue University, USA), Mourad Ouzzani (Purdue University, USA)

Session 3: Privacy and Location Trustworthiness

Privacy-Enabling Abstraction and Obfuscation Techniques for 3D City Models.....53
Martin Kada (University of Stuttgart, Germany), Michael Peter (University of Stuttgart, Germany), Dieter Fritsch (University of Stuttgart, Germany), Oliver Siemoneit (University of Stuttgart, Germany) and Christoph Hubig, (University of Stuttgart, Germany)

Research Issues in Data Provenance for Streaming Environments.....58
Hyo-Sang Lim (Purdue University, USA), Yang-Sae Moon (Kangwon National University, KR) and Elisa Bertino (Purdue University, USA)

On the Impact of Localization Data in Wireless Sensor Networks with Malicious Nodes 63
Mattia Monga (University of Milan, Italy) and Sabrina Sicari (University of Insubria, Italy)

Panel Discussion

Can an Inter-disciplinary Research Community on Location Privacy be Successful?.....71

Yucel Saygin (Sabanci University, Turkey) moderator, Elisa Bertino (Purdue University, USA), Michael Gertz (University of Heidelberg, Germany), Mohamed Mokbel (University of Minnesota, USA), Maria Luisa Damiani (University of Milan, Italy)

Author Index 73

SPRINGL 2009 ORGANIZATION

General Chair
Elisa Bertino (Purdue University, USA)

Program Co-chairs
Maria Luisa Damiani (University of Milan, Italy)
Yücel Saygin (Sabanci University, Turkey)

Web Site Chair
Chenyun Dai (Purdue University, USA)

Proceedings Editor
Ashish Kamra (Purdue University, USA)

Program Committee
Gail-Joon Ahn, Arizona State University, USA
Walid G. Aref, Purdue University, USA
Rafae Bhatti, Oracle, USA
Patrick Capolsini, Université de la Polynésie Française
Reynold Cheng, University of Hong Kong, Hong Kong
Isabelle Cruz, University of Illinois at Chicago, USA
Josep Domingo-Ferrer, Universitat Rovira i Virgili, Catalonia, Spain
Panos Kalnis, KAUST, Saudi Arabia
Paul El Khoury, SAP Research, France
Michael Gertz, University of Heidelberg, Germany
Gabriel Ghinita, Purdue University, USA
James Joshi, University of Pittsburgh, USA
Fosca Giannotti, ISTI-CNR, Italy
Dan Lin, Missouri University of Science and Technology, USA
Ling Liu, Georgia Institute of Technology, USA
Jorge Lobo, IBM, USA
Hervé Martin, University of Grenoble, France
Mohamed Mokbel, University of Minnesota, USA
Cyrus Shahabi, University of Southern California, USA
Bhavani Thuraisngham, University of Texas at Dallas, USA
Kenji Takahashi, NTT, Japan
Tiancheng Li, Purdue University, USA
Vicenç Torra, IIIA-CSIC, Catalonia, Spain

The Role of Security in Scientific Data Management

Michael Gertz
Institute of Computer Science
University of Heidelberg
Heidelberg, Germany
gertz@informatik.uni-heidelberg.de

ABSTRACT

In the past three decades, research and development activities in the area of data security have primarily concentrated on security aspects in traditional domains, such as the business and financial sectors, and, more recently, the medical and health care sectors. Interestingly, compared to major advancements made in these domains, resulting in comprehensive and flexible data security frameworks, there is only little work focusing on security aspects specific to the management of data in natural science domains such as the physical sciences, the life sciences, and the geosciences. Although one can argue that security models, techniques, and architectures developed for the traditional domains can be adopted, scientific data management activities posses some characteristics where important and necessary data security features are either non-existent or are poorly developed. There are several reasons for this. First, scientific data is often complex and go beyond simple record- or XML-based representations, thus requiring sophisticated access control models. A typical example is imagery, which can be found in many such science domains (e.g., GIS-layers in geoscience applications) and for which fine-grained and context-dependent access control is non-trivial. Second, scientific data management often involves complex data life-cycles in which the data undergo transformations and enrichment, especially in the context of data integration. Such life-cycles range from diverse data collection methods to complex computations, often supported by scientific workflows, in which data collected from observations and experiments are combined in support of diverse longitudinal data analysis and exploration tasks. In general, security mechanisms have to be employed at all stages, covering the data, processes, computations, and the archival of the data and data products. All this typically has to occur in a collaborative environment in which data and computation results need to be shared in a flexible and dynamic manner.

We present an overview of the role of security in the context of scientific data management, covering a number of issues related to data security aspects. Using application domains and data usage scenarios from the physical sciences, geosciences and life-sciences, we elaborate on security risks that indicate the need for advanced data security models and techniques in particular in E-Science settings. For complex data processing scenarios, we discuss opportunities and challenges in developing models and techniques in support of authentication models for data and computations researchers can use to verify the correctness and completeness of data and data products, similar to authenticated data publication schemes developed for traditional relational database. We also take a closer look at the inference problem for scientific data, which now can be seen in a new light, given the various ways in which scientific data can be combined and explored in longitudinal data analysis tasks. This aspect holds in particular true for geo-referenced data, which occur in a variety of application domains. Finally, we outline some important security management aspects in the context of the stewardship and preservation of scientific data.

Categories and Subject Descriptors

H.2.7 [**Database Administration**]: Security, integrity, and protection; H.2.8 [**Database Applications**]: Scientific databases; D.4.6 [**Security and Protection**]: Access controls

General Terms

Management, Security

Keywords

access control, data integrity, geosciences, physical sciences, data publication, data inference, scientific workflows

ACM SPRINGL '09, November 3, 2009. Seattle, WA, U.S.A.

Security policies for the Visualization of Geo Data

Patrick Capolsini
Université de la Polynésie française
BP 6570 Faa'a Aéroport
+689 80 38 83
patrick.capolsini@upf.pf

Alban Gabillon
Université de la Polynésie française
BP 6570 Faa'a Aéroport
+689 80 38 80
alban.gabillon@upf.pf

ABSTRACT

We show in this paper that we can use the Or-BAC model [1] to express security policies for the visualization of spatial data. We first add to Or-BAC some visualization predicates and then show how to model various types of spatial data visualization contexts. We finally use these newly defined contexts to write security policies for the visualization of Geo Data.

Categories and Subject Descriptors

K.6.m [**Management of computing and information System**]: Miscellaneous – *Security*

General Terms

Security

Keywords

Access Control, Geo-spatial Data visualization, Or-BAC

1. INTRODUCTION

The core RBAC [2] authorization model considers only static security rules. However, in many applications, there is an increasing need for dynamic security rules. A dynamic security rule can be activated/deactivated depending on some *context*. A context can be a temporal condition, a spatial condition (like the user location), a provisional condition (like the user previous action) etc. Therefore, several extensions to the RBAC model have been proposed in order to cope with contexts: the Generalized Role Based Access Control (GRBAC) [3] incorporates the notion of object role and environment role; in the Context-Role Based Access Control (CRBAC) [4] some constraints should be fulfilled before a permission is assigned to a role; the Or-BAC model [1] allows the security policy designer to express various types of contexts, by using first-order logic. Some models focus on specific contexts, like temporal contexts: the Temporal Role Based Access Control Model (TRBAC) [5] offers means to activate roles periodically; the Generalized TRBAC (GTRBAC) [6] incorporates various temporal constraints on role activation as well as on user-to-role or permission-to-role assignment.

ACM SPRINGL '09 , November 3, 2009. Seattle, WA, USA (c) 2009 ACM ISBN 978-1-60558-853-7/09/11…$10.00

With the growing importance of geographic information in various applications, some important security issues such as hiding sensible areas also arise. These new security challenges involves a new need for dynamic security rules based on some *spatial contexts*. Therefore, security models specifically dealing with spatial contexts have also started to appear: the GeoRBAC [7] model introduces the concept of spatial role to specify spatial condition on user location; the GSAM [8] model introduces the extended concept of geo-temporal role. There are in fact different types of spatial contexts. A spatial context can be the position of a user (in a Location Based System), the zoom level at which a user is looking at a map, the filters and enhancement tools used to visualize geographic objects, the direction followed by a moving object etc.

In [9], we identified and modelled various types of spatial contexts based on the user location and/or the spatial object location. We also showed how to model geo-temporal contexts and contexts related to movement. In this paper, we focus on visualization of geo-data i.e. we show how to model various types of visualization contexts for geo-data and how to express dynamic security rules based on such contexts.

In [9], we also showed that GeoRBAC [10] and GSAM [8] were more or less designed to express security policies based on the user location. They are not suitable to express many of the various visualization contexts for geodata. To our knowledge, the Or-BAC model is the only authorization model which allows the security policy designer to express various types of contexts within a single framework (see [11] and [12]). Or-BAC is a generic security model which formally defines the notion of context and offers a language based on first order logic to specify them. Therefore, we have decided to use Or-BAC for writing our security policies for the visualization of spatial objects.

The remainder of this paper is organized as follows. Section 2 recalls the basic principles of the Or-BAC model. Section 3 defines our geometry model based on the OpenGIS [13] geometry model [14] and introduces some useful visualization predicates. In section 4, we identify several types of visualization contexts and we show how to model them with Or-BAC. In section 5, we describe a real-world visualization application and we give an example of security policy based on these visualization contexts. Section 6 is a comparison of our work with some previous models proposed in the literature. Finally, section 7 concludes this paper and suggests some future works.

2. Or-BAC

In Or-BAC [1], there are eight basic sets of entities: *Org* (a set of organizations), *S* (a set of subjects), *A* (a set of actions), *O* (a set of objects), *R* (a set of roles), *T* (a set of activities), *V* (a set of

views) and *C* (a set of contexts). *Org* $\acute{I}$ *S* (any organization is also a subject) and *S* $\acute{I}$ *O* (any subject is also an object). Subjects, actions and objects are respectively abstracted into *roles*, *activities* and *views*. Roles, activities and views are the *abstract entities* and are always created within the framework of an organization. Abstract entities are organised into hierarchies [15]. Subjects, actions and objects are the *concrete entities*. Each subject (resp. action and object) is linked to one or several roles (resp. activities and views). Abstract entities and concrete entities are linked together by the relations *Empower*, *Use* and *Consider*. *Empower* is a relation over domains *Org*×*S*×*R*. If *org* is an organization, *s* a subject and *r* a role, then *Empower(org,s,r)* means that organization *org* empowers subject *s* in role *r*. *Use* is a relation over domains *Org*×*O*×*V*. If *org* is an organization, *o* an object and *v* a view, then *Use(org,o,v)* means that organization *org* uses object *o* in view *v*. *Consider* is a relation over domains *Org*×*A*×*T*. If *org* is an organization, *a* an action and *t* an activity, then *Consider(org,a,t)* means that *org* considers that action *a* falls within the activity *t*. Any entity in the Or-BAC model may have some *attributes*. This is represented by functions that associate the entity with the value of these attributes. For instance, if *s* is a subject, then *name(s)* represents the value of attribute name of subject s.

A context is ***any kind of constraint*** which may or may not involve the subject and/or the action and/or the object. Organization, subject, object, action and context are linked together by the relation *Hold*. *Hold* is a relation over domains *Org*×*S*×*A*×*O*×*C*. If *org* is an organization, *s* a subject, *a* an action, *o* an object and *c* a context, then *Hold(org,s,a,o,c)* means that within organization *org*, context *c* holds between subject *s*, action *a* and object *o*. For example, a context *Teacher* can be defined as follows:

$$\forall s \in S, \forall a \in A, \forall o \in O, Hold(UPF, s, a, o, Teacher) \leftrightarrow name(o) \in students(s)$$

that is, at organization UPF (University of French Polynesia), context *Teacher* holds between subject *s*, action *a* and object *o* if and only if object *o* is a record corresponding to a student of subject *s*. There is one default context *Default_ctx* which is always true.

In [12], the authors show how to represent different types of contexts with Or-BAC, namely temporal context, user-declared context, prerequisite context and provisional context. They also define some simple spatial contexts using a single built-in predicate *Is_located*.

The *security policy* is specified using the relationships *Permission*, *Obligation* and *Prohibition*. *Permission*, *Obligation* and *Prohibition* are relations over domains *Org*×*R*×*T*×*V*×*C*. If *org* is an organization, *r* a role, *t* an activity, *v* a view and *c* a context then *Permission(org,r,t,v,c)* (resp. *Obligation(org,r,t,v,c)* or *Prohibition(org,r,t,v,c)*) means that in organization *org* role *r* is granted permission (resp. obligation or prohibition) to perform activity *t* on view *v* within context *c*. Instances of *Permission*, *Obligation* and *Prohibition* are called *abstract rules*. These abstract rules are propagated downwards in hierarchies of roles, activities and views through an inheritance mechanism (see [15]).

In Or-BAC, the default security policy is closed which means that anything not explicitly permitted is prohibited. Note that in this paper, for the sake of simplicity, we shall not consider obligations. Indeed, our paper focuses more on how to model spatial contexts than on how to express security rules. For modelling security rules, we apply Or-BAC principles as such. The reader who is interested can refer to [12] where obligations and related concepts of user declared context and provisional context are described in detail. *Concrete rules* are instances of the relationships *Is_permitted*, *Is_prohibited* and *Is_obliged*. *Is_permitted*, *Is_prohibited* and *Is_obliged* are relations over domains *S* × *A* × *O*. Instances of these relationships are logically derived from the abstract rules. The following rule allows us to derive instances of *Is_permitted* from the relation *Permission*:

$$\forall org \in Org, \forall s \in S, \forall o \in O, \forall a \in A, \forall r \in R, \forall v \in V, \forall t \in T, \forall c \in C,$$
$$Permission(org, r, t, v, c) \wedge$$
$$Empower(org, s, r) \wedge Use(org, o, v) \wedge Consider(org, a, t) \wedge$$
$$Hold(org, s, a, o, c) \rightarrow Is\_permitted(s, a, o)$$

that is, if organization *org*, within context *c*, grants role *r* permission to perform activity *t* on view *v* and if *org* empowers subject *s* in role *r* and if *org* uses object *o* in view *v* and if *org* considers that action *a* falls within the activity *t* and if, within *org*, context *c* holds between *s*, *a* and *o* then *s* is permitted to perform *a* on *o*. There is a similar rule for *Is_prohibited* and *Is_obliged*. Specifying a security policy that includes both permissions and prohibitions may lead to conflicts. The Or-BAC model makes the distinction between the *potential conflicts* between abstract rules and the *actual conflicts* between instances of the *Is_permitted* and *Is_prohibited* predicates. The conflict resolution strategy in Or-BAC acts at the abstract level and is based on two complementary approaches : *separation constraints* and *rules priorities*, leading to the concept of prioritized Or-BAC [16] :

- Since a subject can potentially be empowered in different roles, an object can be used in different views, an action can fall within different activities and different contexts can be active simultaneously, every pair of Permission and Prohibition may be potentially conflicting. Such potential conflicts can be eliminated by specifying separation constraints. For instance, if a separation constraint exists between roles r_1 and r_2, then no subject can be empowered in both roles and a Permission assigned to role r_1 cannot get into conflict with a Prohibition assigned to role r_2.
- Remaining conflicts are solved by explicitly assigning priorities to abstract rules.

3. Geospatial data and geometry model

3.1 Geospatial data

GIS software generally deal with two types of spatial data (vector and raster data) but can also accommodate other components such as attribute data (information about an object or feature) and metadata (data about the spatial data). Each geographic data is handled as a layer and layers can be overlaid to get the final desired representation of the dataset. Aerial photography and satellite imagery are common geographically referenced raster data usually used as background layers. Vector layers (see next section for details) can be of different types (points, lines, poly-lines, polygons,…etc) and can represent a large variety of features

themes such as points of interest, roads, buildings, borders of states, elevation contours, restricted access areas and so on. All GIS layers (either raster or vector) can be stored as files or as records in a spatially enabled object-relational database.

The basic visualization features of GIS software (ESRI ArcGIS® for instance) and 3D GeoSpatial viewers (Google earth® for instance) include : zoom in, zoom out, pan, x/y localization, choice of predefined or user-defined scales, layers overlay, hierarchy and re-order of layers, layers transparency or contrast and brightness adjustment.

3.2 Geometric objects

A *georeferenced (geometric) object* is a granule of information that is relevant to an identifiable subset of the Earth's surface [17]. Any geometric object has the following components [18] :

- A *description*. The entity is described by a set of descriptive attributes (e.g. the name of a city)
- A *geometry* which indicates the entity's location and its shape

The geometry model we consider for vector objects is the OpenGIS Geometry Model [14]. In this model, each geometric object belongs to a geometry class. In this paper, we do not consider the whole class hierarchy defined in [14]. For the sake of simplicity, we consider only the branch depicted in figure 1.

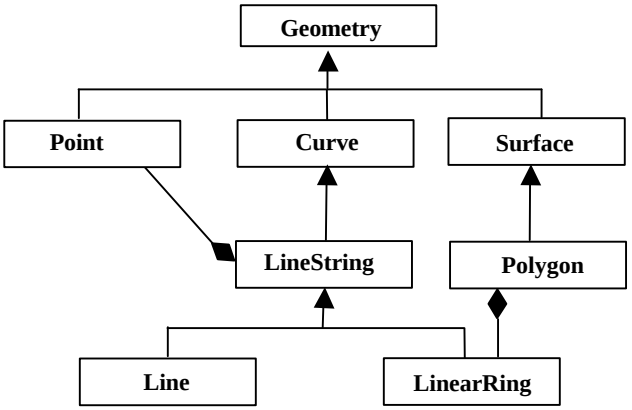


Figure 1: OpenGIS Geometry Class Hierarchy

Let us briefly describe the different classes. For more details, the reader can refer to [14]. `Geometry` is the basis class. It is a non instantiable abstract class. It has some attributes such as `dimension` which indicates the object dimension, `SRID` which contains the Spatial Reference System ID for the geometry object or `geometryType` which indicates the type of the geometric object. It has also some basic methods such as `envelope()` which returns the minimum bounding box (rectangle) enveloping the object. `Point` represents zero-dimensional entities. It has two specific attributes `x` and `y` for its coordinates. `Curve` represents one-dimensional entities. A curve is a sequence of points. The subtype of `Curve` specifies the form of the interpolation between points. It has two specific attributes `startpoint` and `endpoint`. It has one subclass `Linestring` which uses linear interpolation between points. `Linestring` has a method `pointN(N:integer)` which returns the specified point `N` in the linestring. `Line` is a linestring with exactly two points. LinearRing is a linestring that is closed and simple (a linestring is simple if it does not pass through the same point twice with the possible exception of the two end points). Note that a particular case of linearRing is the bounding box. `Surface` represents two-dimensional entities. A surface has one exterior boundary and 0 or more interior boundaries defining some "holes" in the surface. A polygon is a planar surface. `Polygon` has two methods `exteriorRing()` which returns the lineRing defining the exterior boundary and `interiorRingN(N:integer)` which returns the specified interior lineRing `N` in the polygon.

In section 2, we defined the set of objects O and the set of subjects S, with $S \subset O$ (any subject is also an object). We assume all objects in O to be geo-referenced. Therefore any object has some descriptive attributes and some spatial attributes and methods. These spatial attributes and methods can be used for specifying contexts, like any other attributes. For example, if p is an object whose geometry is a point then *x(p)* and *y(p)* represents its coordinates. If l is a Linestring then *pointN(l,3)* represents the third point of l…etc. If talking about the geometry of an entity is irrelevant then the geometry of this entity is the empty geometry $\varnothing$. Since subjects are mostly users, the geometry of subjects is generally a point. However, it could also be a polygon if the exact subject position cannot be determined precisely or should not be disclosed for privacy reasons. Location and/or shape of any object may change over time. This is obviously true for users whose coordinates are updated in real-time (thanks to GPS devices for example), but it can also be true for any other object.

3.3 Visualization predicates

Visualization predicates are used to test for a specific property between two geographic objects within the scope of a display-like operation.

We define two predicates: *Overlay* and *StrictOverlay*. We consider these predicates to be built-in Or-BAC predicates. They are both defined over domain $O \times O$. Let g_1 and g_2 be two geographic objects involved in a visualization operation. We first define *Overlay* as follows:

- *Overlay*(g_1, g_2) means that geographic object g_1 lays over geographic object g_2

This predicate means that g_1 can be any geographic layer located either directly on top of g_2 or above any layer itself above g_2.

To express that a geographic object is overlaid directly on another one, we define predicate *StrictOverlay* as follows:

- $\forall g_1, \forall g_2,$
  $StrictOverlay(g_1, g_2) \leftrightarrow Overlay(g_1, g_2) \wedge \neg(\exists g_3, Overlay(g_3, g_2) \wedge Overlay(g_1, g_3))$

4. Using Or-BAC to model contexts related to visualization of spatial data

In this section we try to figure out various types of visualization contexts (without pretending to be exhaustive) and we show how to model them using Or-BAC. Table 1 summarizes some examples of visualization contexts and indicates whether the context applies only to raster data or if it applies to both raster and

vector geographic objects. A reference to the concerned sub-section is also given to get more details.

Type	Definition	Security rule	Geo data	Sect.
Zoom-in factor	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O, \forall z \geq 1,$ $Hold(org, s, a, o, mzf(z)) \leftrightarrow$ $scale(a) \leq (z \times Defaultscale)$	*Permission(UPF, Employees, Display, Aerial_SatelliteView, mzf(2))*	Vector & Raster	4.1
Simple Overlay	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O,$ $Hold(org, s, a, o, LayOverTiahuraMPA)$ $\leftrightarrow Overlay(o, Tiahura\_MPA)$	*Permission(UPF, Researchers, Display, ReefMonitoringView, LayOverTiahuraMPA)*	Vector & Raster	4.2.1
Strict Simple Overlay	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O,$ $Hold(org, s, a, o, StrictLayOverTiahuraMPA)$ $\leftrightarrow StrictOverlay(o, Tiahura\_MPA)$	*Permission(UPF, Researchers, Display, ReefMonitoringView, StrictLayOverTiahuraMPA)*	Vector & Raster	4.2.2
Composed Simple Overlay	Composition of Simple Overlaying contexts defined using operators AND ($\&$), or ($\oplus$) and NOT ($\neg$)	*Permission(UPF, Researchers, Display, ReefMonitoringView, LayOverTiahuraMPA* & LayOver*LagoonAerial)*	Vector & Raster	4.2.3
Multiple Overlay	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O,$ $Hold(org, s, a, o,$ $Monitoring\_over\_MPA\_over\_Lagoon) \leftrightarrow$ $Overlay(Monitoring\_points, MPA\_zones)$ $\wedge\ Overlay(MPA\_zones, LagoonAerial)$	*Permission(UPF, Researchers, Display, SpatialObjectsView, Monitoring_over_MPA_over_Lagoon)*	Vector & Raster	4.2.4
Strict Multiple Overlay	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O,$ $Hold(org, s, a, o,$ $Strict\_Monitoring\_over\_MPA\_over$ $\_Lagoon) \leftrightarrow$ $StrictOverlay(Monitoring\_points,$ $MPA\_zones) \wedge$ $StrictOverlay(MPA\_zones, LagoonAerial)$	*Permission(UPF, Students, Display, SpatialObjectsView, Strict_Monitoring_over_MPA_over_Lagoon)*	Vector & Raster	4.2.5
On top Overlay	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o_1 \in O,$ $Hold(org, s, a, o_1, Lay\text{On}Top) \leftrightarrow$ $\neg(\exists o \in O, Overlay(o, o_1))$	*Permission(UPF, Users, Display, OutliningView, LayOnTop)*	Vector & Raster	4.2.6
Background Overlay	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o_1 \in O,$ $Hold(org, s, a, o_1, LayOnBackground) \leftrightarrow$ $\neg(\exists o \in O, Overlay(o_1, o))$	*Permission(UPF, Users, Display, OutliningView, LayOnTop)*	Vector & Raster	4.2.7

Transparency	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O,$ $\forall z_1 \in [0,1], \forall z_2 \in [0,1],$ $Hold(org, s, a, o, transpinterval(z_1, z_2)) \leftrightarrow$ $transparency(o) \geq z_1 \wedge$ $transparency(o) \leq z_2$	*Permission(UPF, Researchers, Display, MPAView , transpinterval(0, 0.5) & LayOverLagoonAerial)*	Vector & Raster	4.3
Contrast & Brightness	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O,$ $\forall z_1 \in [-1,1], \forall z_2 \in [-1,1],$ $Hold(org, s, a, o, contrastinterval(z_1, z_2)) \leftrightarrow$ $contrast(o) \geq z_1 \wedge contrast(o) \leq z_2$	*Permission(UPF, Researchers, Display, Aerial_SatelliteView, contrastinterval(-1, 0.5))*	Raster	4.4
	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O,$ $\forall z_1 \in [-1,1], \forall z_2 \in [-1,1],$ $Hold(org, s, a, o, brightnessinterval(z_1, z_2)) \leftrightarrow$ $brightness(o) \geq z_1 \wedge brightness(o) \leq z_2$	*Permission(UPF, Researchers, Display, Aerial_SatelliteView, brightnessinterval(-0.5, 0.5))*		
Filters	$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O, \forall z \in [0,1],$ $Hold(org, s, a, o, Linear\_stretch(z)) \leftrightarrow$ $linear\_stretch(o) = z$	*Permission(UPF, Researchers, Display, Aerial_SatelliteView, Linear_stretch(0.02))*	Raster	4.4

Table 1 : Examples of visualization contexts

4.1 Zoom-in factor

In some spatially aware access control models like GSAM, *zoom-in* is considered as a separate privilege like read or write. We prefer to model the zoom-in operation as a context of another operation, like the *display* operation for instance. We believe that this approach facilitates the interpretation of the security rules and avoids conflicts. For instance, if display and zoom-in were two distinct privileges, a conflict could arise between a rule stating that a geographic object can be zoomed-in and another rule prohibiting the display of the same object.

Let *Defaultscale* be the default scale at which a geometric object is displayed. Since defining a function returning a context is possible with Or-BAC (see [11] for the definition of functions *before_time* and *after_time* for instance). In table 1, we define function *mzf* (maximum zoom-in factor) that takes as input a zoom-in factor and returns a spatial context. That is, at organization *org*, context *mzf(z)* holds between subject *s*, action *a* and object *o* if and only if the *scale* parameter of action *a* is less than or equal to the default scale *Defaultscale* multiplied by the zoom-in factor *z*. Let us note that if action *a* does not have a scale parameter (descriptive attribute) then function *mzf* will never return any context. The given example of permission states that employees at UPF have the permission to display aerial or satellite photographs of the lagoon with a maximum zoom-in factor of 2.

4.2 Overlaying

In the field of electronic mapping, overlaying geographic data is of main importance to visualize what is called "an electronic map". Each geographic object (see section 3) is handled as a layer and layers are overlaid to get the final desired representation of the dataset. For instance, geographic objects like state boundaries, roads or regions of interest are overlaid on a background image like an aerial photography or a satellite image. Being able to control which geographic object can be overlaid on which other geographic object can be a security requirement. Indeed, since each layer is displayed according to a transparency parameter, the order in which the layers are displayed can hide/reveal some information. In this subsection we try to figure out a typology of overlaying contexts and show how to model them.

4.2.1 Simple overlay

In the given example (see Table 1), we consider *Tiahura_MPA* to be a geometric object whose geometry is a polygon corresponding to the Marine Protected Area located around the *Tiahura* islet. We use the *Overlay* predicate defined in section 3 to define a context called *LayOverTiahuraMPA* (lay over "Tiahura" MPA) which holds between subject *s*, action *a* and object *o* if and only if object *o* lays (strictly or not) over object *Tiahura_MPA*. The permission rule grants researchers at UPF the permission to display any object belonging to the *ReefMonitoringView* view if and only if it overlays the *Tiahura* MPA geographic object.

4.2.2 Strict simple overlay

Considering the same geographic object *Tiahura_MPA*, we can use the *StrictOverlay* predicate to define a new context called *StrictLayOverTiahuraMPA* (Strictly lay over "Tiahura" MPA) which holds between subject *s*, action *a* and object *o* if and only if object *o* is the very first object laying over object *Tiahura_MPA*.

4.2.3 Composed simple overlay

Since the Or-BAC model allows the composition of contexts (see [11]) using Boolean operators AND ($\&$), or ($\oplus$) and NOT ($\neg$), the security policy may include rules involving more than one context. This possibility can be used to define multiple overlaying contexts.

Table 1 defines context LayOver*LagoonAerial* in the same way we defined *LayOverTiahuraMPA* but using object *LagoonAerial* (an aerial photography of the lagoon) instead of object *Tiahura_MPA*. It becomes then possible to compose these two contexts to express a new rule granting the permission to display any geographic object on top of these two geographic objects. The provided example of permission grants researchers at UPF the permission to display any object *o* belonging to the *ReefMonitoringView* view provided object *o* overlays both the *Tiahura* MPA and the aerial photography of the lagoon. Notice that, (i) any other geographic object can be interposed between object *o* and objects *Tiahura_MPA* and *LagoonAerial* (ii) there is no overlaying order between objects *Tiahura_MPA* and *LagoonAerial*.

4.2.4 Multiple overlay

The example given in table 1 uses the *Overlay* predicate to define a context *Monitoring_over_MPA_over_Lagoon* saying that objects *Monitoring_points* (some monitoring points where scientist conduct regular biological survey), *MPA_zones* (Marine Protected Areas all around the island) and *LagoonAerial* (aerial photography of the lagoon) are displayed in a given order. However, these three objects are not strictly ordered, i.e. other layers can be interposed between them.

The corresponding rule grants researchers at UPF the permission to display any spatial object if and only if objects *Monitoring_points*, *MPA_zones* and *LagoonAerial* are *also* displayed in the order given in context *Monitoring_over_MPA_over_Lagoon*.

4.2.5 Strict Multiple overlay

We define the S*trict_Monitoring_over_MPA_over_Lagoon* context in the same manner as the previous one except that the three objects are strictly overlaid on top of each other.

4.2.6 On top Overlay

Still using the *Overlay* predicate, we can define a context saying that a given geographic object has no other geographic object laid over it.

The rule given in table 1 grants every user at UPF the permission to display any spatial object belonging to view *OutliningView* (User-defined points, lines or polygons) if and only if this spatial object is displayed as the topmost object of the map.

4.2.7 Background Overlay

In the same way, it is possible to define a context saying that a given geographic object lays at the background of a map. The corresponding rule grants every user at UPF the permission to display any spatial object belonging to view *Aerial_SatelliteView* (aerial or satellite photographs) if and only if this photography is displayed as the background object of the map.

4.3 Transparency

In addition to the Overlay possibility, any GIS software provides the ability to adjust the display transparency of objects. Indeed, as already mentioned, geographic objects are overlaid on each other, therefore some geographic objects (a colored polygon for instance) may hide other objects lying "under". To gain display of objects located "under", there are two options: either changing the overlay order but this may possibly hide/reveal other areas or adjusting the transparency (from 0% to 100%) of the upper object allowing then the simultaneous visualization of different objects. Furthermore, denying the possibility to change the transparency of an object is a mean of hiding the information lying under it.

We model *transparency* as a parameter of geographic objects and we define context *transpinterval*(z_1, z_2) (transparency interval) which holds between subject *s*, action *a* and object *o* if and only if the *transparency* parameter of object *o* is within interval [z_1, z_2].

For instance, if we compose the *transpinterval* context and the *LayOverLagoonAerial* context defined in table 1, we can grant researchers the permission to adjust, within a given interval, the transparency of the polygons defining any Marine Protected Area object when this object is overlaid on the aerial photography of the lagoon. This rule makes the assumption that with a transparency above 50% some sensitive details on the aerial photography would be revealed.

4.4 Contrast Brightness and Filters

While transparency applies to all types of data, adjusting the contrast and/or brightness of selected geographic objects is a common operation used, on raster data, to highlight certain characteristics of the geographic object. For instance, a contrast setting well suited could reveal some hidden details of an aerial photography. Contrast and brightness are usually expressed on a scale from -100% to 100%. We also model *contrast* and *brightness* as parameters of geographic objects and use these parameters to define contexts *constrastinterval*(z_1, z_2) (contrast interval) and *brightnessinterval*(z_1, z_2) (brightness interval).

There is a large amount of filter operations that may be applied to raster images, including linear, Laplacian, Gaussian, high/low pass, median, texture, adaptive and so on. We model filters as we model contrast and brightness that is, as parameters of geographic objects. We use these parameters to define contexts. For instance, we can define a context *Linear_stretch(z)* which applies a linear stretch filter with a z % clip on both ends of the displayed.

5. Example of a security policy

In this section, we consider an organization called *Urbanism Service (UrbSv)* which is responsible of collecting and updating all geographic data over French Polynesia. This service distinguishes two types of users of geographic data: users having public access only and authorized users belonging to the Urbanism service personnel. *UrbSv* aims at offering to all its users an online GIS platform to display some data of interest concerning some of the islands of French Polynesia. The data set includes some aerial or satellite images, all the zoning of the existing Marine Management Plans (MMP) or Urban

Development Plans (UDP) as well as the location of some fragile and expansive equipments like tide gauges, permanent GPS stations or communication devices like satellite antennas. Additionally, a visualization prohibition applies on some given military zones (barracks, military airports or former nuclear test sites) which must always be hidden by means of geographic objects of type polygon if aerial or satellite imagery is displayed. Figure 2 shows the relationships between abstract entities and concrete entities in the Or-BAC model.

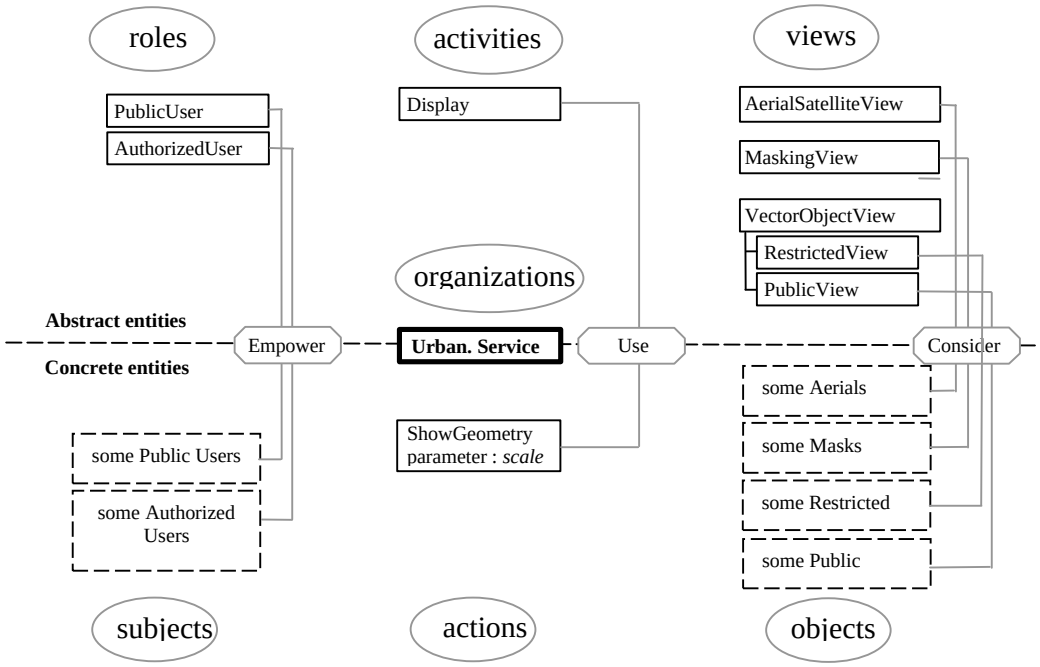


Figure 2: Synopsis of our example

Before defining the security policy, we need to define the organization, the roles, the activities, the views, the subjects, the actions, the objects and the contexts:

- Organization: *Urbanism Service (UrbSv)*
- Roles:
  - *User* with two sub-roles: *PublicUser* and *AuthorizedUser*.
  - We assume *separated_role(UrbSV, PublicUser, AuthorizedUser)* i.e. we assume a subject cannot be empowered in both roles
- Activities: *Display* is the only activity
- Views:
  - *Aerial_SatelliteView*
  - *MaskingView*
  - *VectorObjectView* with two sub-views:
    - *RestrictedView*
    - *PublicView*
  - We assume that all views are mutually separated using the *separated_view* function of the Or-BAC model. For instance: *separated_view(UrbSv, Aerial_SatelliteView, MaskingView)*
- Subjects: Several public and authorized users
- Actions: *ShowGeometry* (implements activity *Display*): for graphically displaying geometric objects. *ShowGeometry* has one parameter *scale* indicating the scale at which objects are displayed.
- Objects:
  - Various objects of type polygon (implementing view *MaskingView)* masking some military barracks or other sensitive infrastructures
  - Several objects belonging to the defined views.
- Constants: *DefaultScale* giving the default scale at which a geometric object is displayed.

The following contexts are also defined:

- Context ***mzf*** defined in section 4 but redefined here in the scope of organization *UrbSv* as follows:

$$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O, \forall z \geq 1,$$
$$Hold(UrbSv, s, a, o, mzf(z)) \leftrightarrow$$
$$scale(a) \leq (z \times Defaultscale)$$

- Context ***Masked_Military_zones*** expressing that all geographic objects belonging to view *MaskingView* must be overlaid on object o_1. Using first order logic language, this translated by: "whatever the object o_1, it is not possible to find an object o_2 belonging to view *MaskingView* which is not overlaid on object o_1".

$$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o_1 \in O,$$
$$Hold(UrbSv, s, a, o_1, Masked\_Military\_zones) \leftrightarrow$$
$$\neg(\exists o_2, Use(o_2, MaskingView) \wedge \neg Overlay(o_2, o_1))$$

- Contexts ***transpinterval(z1, z2)*** and ***brightnessinterval(z1, z2)*** already defined in section 4 but redefined here in the scope of organization *UrbSv* as follows:

$$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O, \forall z_1, z_2$$
$$\forall z_1 \in [0,1], \forall z_2 \in [0,1],$$
$$Hold(UrbSv, s, a, o, transpinterval(z_1, z_2)) \leftrightarrow$$
$$transparency(o) \geq z_1 \wedge transparency(o) \leq z_2$$
$$\forall org \in Org, \forall s \in S, \forall a \in A, \forall o \in O,$$
$$\forall z_1 \in [-1,1], \forall z_2 \in [-1,1],$$
$$Hold(UrbSv, s, a, o, brightnessinterval(z_1, z_2)) \leftrightarrow$$
$$brightness(o) \geq z_1 \wedge brightness(o) \leq z_2$$

We can now express our security policy directly inspired by a concrete-use case. Basically, the security policy expresses the fact that: (i) aerial or satellite imagery can be displayed if some precise military zones are masked and (ii) authorized users have the right to display any geographic object while general public users have some restrictions on displayable objects as well as on potential filtering.

Rule 1: All users are granted permission to display any aerial photography or satellite image (view *Aerial_SatelliteView*) with a maximum zoom-in factor of 10 if and only if object *Military_zones* overlay all other objects. Let us notice that, since nothing is said about contrast, brightness or filtering, users are not restricted to use these operations.

$$Permission(UrbSv, User, Display, Aerial\_SatelliteView, Masked\_Military\_zones \ \& \ mzf(10))$$

Rule 2: Users are permitted to display any objects of view *MaskingView* with a transparency only equal to 0% (fully opaque)

$$Permission(UrbSv, User, Display, MaskingView, transpinterval(0,0))$$

These first two rules ensure that all users are granted permission to display any aerial photography or satellite image provided the zoom-in factor is not greater then 10 and all the masks are also displayed.

Rule 3: Authorized users are granted permission to display any vector object:

$$Permission(UrbSv, AuthorizedUser, Display, VectorObjectView)$$

Rule 4: Public users are granted permission to display public vector data:

$$Permission(UrbSv, PublicUser, Display, PublicView)$$

Rule 5: Public users are prohibited to display aerial or satellite imagery with a brightness interval outside [–100%, +20%]

$$Prohibition(UrbSv, PublicUser, Display, Aerial\_SatelliteView, \overline{brightnessinterval(-1,0.2)})$$

Regarding the above security policy, we can make the following comments addressing conflicts between rules:

Thanks to the separation constraints defined above (e.g. a subject cannot be empowered in both roles *PublicUser* and *AuthorizedUser* and all views are mutually exclusive), we avoid many potential conflicts. However, a potential conflict still remains between rules 1 and 5 if the user is a *PublicUser*, the object belongs to view *Aerial_SatelliteView*, all the masks are displayed, the zoom-in factor is not greater than 10 and the requested brightness is outside [-100%, 20%]. We solve this conflict by assigning to rule 5 a priority which is higher than the priority of rule 1.

6. Related works

During the last decade there has been a significant increase in the amount of papers dealing with access control to geographic data. Among all these papers we distinguish between papers dealing with the location of the user (Location Based Systems or LBS) and/or geographic objects and papers related to the access and visualization of geo-data.

The first category includes SRBAC [19] and geo-RBAC[10]. In the SRBAC [19] model, roles are dynamic in the sense that a role is assigned different sets of permissions according to different geographical areas. Therefore, the permissions granted to a user assigned to a particular role will differ according to the location of the user. In SRBAC the space model is very simple and targeted to wireless networks applications. The whole space is divided into adjacent cells and the logical position of a user is simply defined as a set of contiguous cells. Such a space structure is very rigid. The spatial granularity is fixed and does not allow any refinement regarding the position of the user. The concept of dynamic role reduces the number of named roles within the security policy but it is at the cost of a poor legibility of roles since a named role hides different actual roles. The Geo-RBAC model was introduced for the first time in [7] then further formalized and extended in [10]. In this model, spatial entities are used to model objects, user positions and geographically bounded roles. Geo-RBAC defines the concept of *spatial role*. A spatial role is a pair <*r*,*e*> where *r* is a traditional role and *e* the spatial extent of the role. The role extent defines the boundaries of the space in which the role can be assumed by the user. Users have both a physical position (given by a GPS device for example) and a logical position (the road, the building, the region they are located in). There is a function mapping the logical position of a user to his/her real position. During a session, a user must be logically located within the spatial extent of a given role to activate this role. Spatial objects, called *features* according to the OGC terminology, are grouped into *feature types* (for instance Road, Town or Car). The authors also define the concept of *role schema*. Authorizations can be either assigned to the role schema (and are consequently inherited by all role instances of the schema,) or directly to the role instances. Finally, hierarchical Geo-RBAC (Geo-HRBAC) adds to Geo-RBAC the possibility to organize spatial roles and role schemas into hierarchies.

In the second category of papers, we find [20], [21] as well as the GSAM model described in [22], [23], [24] and [8]. In [20] the

authors aim at controlling the way vector-based data are accessed through a Web Map Management Service by users having different profiles. In this model, an authorization object is a spatial feature class (a set of instances or features) and authorizations on a given feature class states which instances of the feature class can be accessed, in which way and by which roles. In addition, authorizations are assigned a geographical scope called "window" (a polygon) indicating the portion of territory to which the authorization applies. Regarding the visualization of geographic data, this model has two main drawbacks: prohibitions are not considered and authorisation rules for objects at a finer level of granularity (on single features for example or on feature class attributes) cannot be expressed. In [21], authors allow the specification of rules controlling the access to complex structured spatial data stored in spatial databases. Although this model also supports the concept of authorization window to specify the region on which the authorization applies, it differs quite significantly from the model in [20]. Indeed, this model: (i) uses a more complex map model admitting either a geometrical or topological representation of the same spatial object; (ii) supports different propagation rules making access control more flexible; (iii) also supports both positive and negative authorizations. However, the model does not support roles and does not allow to define rules involving mobile subjects. The GeoSpatial Authorization Model (GSAM) is probably the best access control authorization model for the visualization of geographic data. GSAM was one of the first models dealing with access control systems for geographical data and was first proposed in [22]. The model deals with remote sensing imagery and focuses on sensible information revealed by high resolution satellite imagery. Objects are either raster (satellite or airborne) or vector (points, lines, polygons, …) images. Each object (an image) has attributes defining the spatial extend of the image, its resolution and a timestamp representing either the download date (raster images) or the last update (vector images). Additionally, tabular data may be associated to vector images. In addition to the usual privileges such as *insert*, *delete* and *update*, GSAM also supports actions required to provide controlled access to multi-resolution imagery, namely *view*, *zoom-in*, *identify* and *overlay*. A GSAM authorization is a triple <*sub, obj, pr*> where *sub* is a subject, *obj* is an object, a set of objects or a rectangular geographical region and *pr* is a privilege or a set of privileges. The model is primarily used to restrict the ability to zoom-in on images or portions of images. The original GSAM model was extended in [23] in three ways. Roles were introduced in order to abstract the concept of *subject*, defining negative authorizations (ie. prohibitions) became possible and finally authorizations were extended with a new *condition* attribute containing a condition expressed over subjects and object attributes. In [24], the authors add the possibility to define some basic temporal constraints on both subjects and geospatial objects. Finally, in [8], the authors introduce the concept of *geo-temporal role*. A geo-temporal role is a pair <*r*,*sc*> where *r* is a traditional role and *sc* a *scene*. The concept of scene resembles the concept of spatial extent in Geo-RBAC except that a scene includes a temporal extent.

We believe that the Or-BAC model has many advantages over this last model. Indeed, although GSAM borrows the concept of role from the traditional RBAC model, it is an *ad hoc* model more or less designed for a particular type of geospatial application namely for regulating access to satellite images. Consequently, GSAM has not the flexibility of the Or-BAC model, which is a generic authorisation model. Furthermore, the Or-BAC model including our visualization primitives inherits the qualities of the core Or-BAC model. For instance, we benefit from the useful and innovative concepts of activity and view for structuring the security policy. We also benefit from the conflict resolution strategy of the Or-BAC model. Conflict resolution is barely addressed in GSAM and some other models whereas in [16], the authors show that conflict detection in Or-BAC is tractable in polynomial time.

7. Conclusion and perspectives

In this paper we showed how to model security rules for the visualization of spatial data. We only extended the core Or-BAC first order logic language with two simple predicates *Overlay* and *StrictOverlay* and introduced a typology of contexts based on these new predicates as well as on a visualization parameter of activity *Display* (scale parameter) and some attributes of geographic objects like transparency or brightness. Through a real life application, we showed how we can easily express various kinds of security rules applying to geo data visualization. Finally, we showed that many visualization contexts cannot be easily handled by existing spatial models like GSAM. Future works shall include:

- Designers of the Or-BAC model have developed MotOrBAC [25]. MotOrBAC is a security policy tool which can be used to specify, simulate and administrate security policies. MotOrBAC has been developed on top of the Or-BAC application programming interface (API), a java API. MotOrBAC uses the Jena inference engine [26] for deriving conflicts and concrete rules from abstract rules. We plan to extend MotOrBAC with the proposed predicates in order to simulate spatial visualization policies.
- In this paper, we limited ourselves to a two dimensional geometric model, future works could also consider a three dimensional geometric model.
- A complete implementation of an enforcement architecture is also a great challenge but requires a huge amount of human resources that we do not currently have.

Acknowledgments

This work was conducted as part of the ANR funded project under reference ANR-SESUR-2007-FLUOR.

References

[1] El-Kalam, A., et al. *Organization Based Access Control*. in *4th IEEE International Workshop on Policies for Distributed Systems and Networks (Policy'03)*. 2003. Como, Italy: IEEE.

[2] Sandhu, R., et al., *Role-based access control models.* IEEE Computer, 1996. **29**(2): p. 38-47.

[3] Moyer, M. and M. Ahamad, *Generalized Role-Based Access Control*, in *Proceedings of the 21st International Conference on Distributed Computing Systems*. 2001, IEEE Computer Society.

[4] Park, S.-H., Y.-J. Han, and T.-M. Chung. *Context-Role Based Access Control for Context-Aware Application*. in *High Performance Computing and Communications*. 2006. Munich, Germany: Springer.

[5] Bertino, E., P.A. Bonatti, and E. Ferrari, *TRBAC : A temporal role-based access control model.* ACM Transactions on Information and System Security (TISSEC'01), 2001. **4**(3): p. 191-233.

[6] Joshi, J.B.D., et al., *A generalized temporal role-based access control model.* IEEE Transactions on Knowledge and Data Engineering, 2005. **17**(1): p. 4-23.

[7] Bertino, E., et al. *GEO-RBAC : A spatially Aware RBAC*. in *ACM Symposium on Access Control Models and Technologies (SACMAT'05)*. 2005. Stockholm, Sweeden.

[8] Atluri, V. and S.A. Chun, *A geotemporal role-based authorization system.* International Journal of Information and Computer Security, 2007. **1**(1/2): p. 143-168.

[9] Gabillon, A. and P. Capolsini. *Dynamic Security rules for Geo Data*. in *International workshop on Autonomous and Spontaneous Security (SETOP'09)*. 2009. St Malo, France: Springer-Verlag.

[10] Damiani, M.L., et al., *GEO-RBAC : A spatially Aware RBAC.* ACM Transactions on Information Systems and Security, 2006. **00**(00): p. 1-34.

[11] Cuppens, F. and A. Miège. *Modeling Contexts in the Or-BAC Model*. in *19th Annual Computer Security Applications Conference (ACSAC '03)*. 2003. Las Vegas, NV, USA.

[12] Cuppens, F. and N. Cuppens-Boulahia, *Modeling Contextual security policies.* International Journal of Information Security (IJIS'08), 2008. **7**(4): p. 285-305.

[13] OGC. *Open Geospatial Consortium Inc. - About Us*. 2008 [cited; Available from: http://www.opengeospatial.org/about.

[14] Herring, J.R., *OpenGIS(R) Implementation Specification for Geographic information - Simple feature access - Part 1 : Common architecture.* Open Geospatial Consortium Inc., 2006. OGC(R) 06-103r3.

[15] Cuppens, F., N. Cuppens-Boulahia, and A. Miège. *Inheritance hierarchies in the Or-BAC model and application in a network environment*. in *Second Foundation of Computer Security WorkShop (FCS'04)*. 2004. Turku, Finland.

[16] Cuppens, F., N. Cuppens-Boulahia, and M.B. Ghorbel, *High level conflict management strategies in advanced access control models*, in *WorkShop on Information and Computer Security (ICS'06)*. 2006, Elsevier Sciences: Timisoara, Roumania.

[17] Janée, G., J. Frew, and L.L. Hill, *Issues in Geo-referenced Digital Libraries*, in *D-Lib Magazine*. 2004.

[18] Rigaux, P., M. Scholl, and A. Voisard, *Spatial Databases with application to GIS*. 2002: Elsevier. 410.

[19] Hansen, F. and V. Oleshchuk. *SRBAC : A spatial Role-Based Access Control Model for Mobile Systems*. in *7th Nordic workshop on secure IT systems (NORDSEC'03)*. 2003. Gjvik, Norway.

[20] Bertino, E., M.L. Damiani, and D. Momini. *An access control system for a Web map management service*. in *In Proceedings of the 14th International Workshop on Research Issues on Data Engineering: Web Services for e-Commerce and e-Government Applications (RIDE'04)*. . 2004.

[21] Belussi, A., et al. *An Authorization model for geographical maps.* in *In Proceedings of the 12th annual ACM International Workshop on Geographic Information Systems (RIDE'04)*. . 2004. Washington DC, USA.

[22] Chun, S.A. and V. Atluri. *Protecting privacy from continuous high-resolution satellite surveillance*. in *In Proceedings of the 14th IFIP 11.3 Annual Working Conference on Database Security*. 2000. Schoorl, The Netherlands.

[23] Atluri, V. and P. Mazzoleni. *A uniform indexing scheme for geo-spatial data and authorizations*. in *In Proceedings of the 16th IFIP WG 11.3 Conference on Data and Application Security*. 2002.

[24] Atluri, V. and S.A. Chun, *An authorization Model for Geospatial Data.* IEEE Transactions on Dependable and Secure Computing, 2004. **1**(4): p. 238-254.[25] Autrel, F., et al., *MotOrBAC 2: a security policy tool*, in *3rd Conference on Security in Network Architectures and Information Systems (SAR-SSI'08)*. 2008: Loctudy, France. p. 273-288.

[26] Carroll, J.J., et al., *Jena : Implementing the semantic web recommendations*, in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*. 2004: New York, NY, USA. p. 74-83.

A Generalized Context-based Access Control Model for Pervasive Environments

José Bringel Filho*
University of Grenoble
LIG Laboratory, STEAMER team
681, rue de la Passerelle
38402, Saint Martin d'Hères, France
bringel@imag.fr

Hervé Martin
University of Grenoble
LIG Laboratory, STEAMER team
681, rue de la Passerelle
38402, Saint Martin d'Hères, France
herve.martin@imag.fr

ABSTRACT

Pervasive Computing Environments enable new opportunities for users to share and to access resources anytime and anywhere in a more natural way, making access control a critical issue. These heterogeneous and dynamic sensor-rich environments characterized by frequent and unpredictable changes on user's, resource's, and environment situations, call for access control solutions that allow dynamically adjust access permissions based on information describing the conditions of these entities (context), such as location and time. Some research attempts have been done based on existing models, which context information is used as an optional attribute for limiting the scope of access control permissions. However, these approaches normally exploit identities and roles dynamically assigned to the users in order to grant access permissions, which is an inappropriate solution for open and dynamic environments which we cannot assume the existence of predefined roles and user-role associations. In this scenario, we claim that access permissions should be assigned to the users only based on context information characterizing the three most important entities of any access control framework: owners, requestors, and resources. Thus, this paper proposes a generalized context-based access control model for making access control decisions completely based on context information.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; H.3.4 [**Systems and Software**]: Distributed systems

General Terms

Security, Distributed systems

*Supported by the Programme AlBan, the European Union Programme of High Level Scholarships for Latin America, scholarship no. E06D104158BR.

ACM SPRINGL '09, November 3, 2009. Seattle, WA, USA

Keywords

Access control, pervasive computing, context-based decisions

1. INTRODUCTION

Pervasive Computing Environments (PCE) offer to the users the possibility for access, to create and to share resources, from anywhere at anytime, interacting dynamically with other surrounding devices and users. The constant advancement of sensor-rich mobile devices (e.g. GPS, Bluetooth, accelerometer) has enabled to the users for creating dynamically content (e.g. photos, video, micro-blog), which can be annotated at creation time with information characterizing the situation (e.g. location, time, nearby devices)[17].

However, these sensor-rich environments introduce new access control requirements that cannot be solved by usual existing models such as MAC (Mandatory Access Control Model) [1] and RBAC (Role-Based Access Control Model) [14]. They were initially specified for closed and relatively unchangeable distributed systems that deal only with a set of known users who access a set of known services/resources. Furthermore, they do not take into account information characterizing the situation of owners, requestors, and resources, such as location and time, when determining whether access should be allowed or not to the users [8].

In systems implementing RBAC-based access control approaches, each user should be associated with one or more roles in order to have permissions on resources. By contrast, in PCE we cannot assume that access control frameworks know previously all users and that it is able to apply user-role associations. For instance, in the scenario of an international workshop in which the collaboration relationships between the participants are dynamic and unpredictable, we cannot determine previously the persons that will interact with us and possibly share/access some resource (e.g. presentation file). For granting access on resources using a RBAC-based approach, the administrator or the resource owner in a user-centric access control framework, should create an identity for each participant assigning to it a role (e.g. visitor), which will inherit role-associated permissions. However, this administration task could be made simpler if access permissions are granted to the users only based on information characterizing their situations (e.g. in the workshop), such as location (e.g. the conference room) and time (e.g. at session time).

In fact, the mobility of devices and users causes continuously changes in the situation (i.e. context) of these entities,

as well as in the interactions among them. Dey et. *al* [4] define context *as any information that can be used to characterize the situation of an entity or object relevant to the interaction between a user and an application.* From access control point of view, context information can characterize the situation of three main elements present in any access control system: owners, requestors, and resources. Granting access on resource to the users without taking into account the current context associated with these entities could compromise the security in PCE.

Some research [2, 12, 11, 13] attempts have been done under context-aware access control models, which context information is an optional attribute used in order to limit the scope of access control policies. For instance, CRBAC [12], GRBAC [11], and TRBAC [2] extended the RBAC model in order to take into account some context dimensions for restricting access permissions on resources. However, as with RBAC these models are not suitable to control access on resources in open and dynamic environments with frequent and unpredictable changes, which we cannot assume the existence of predefined roles and user-role associations.

In order to overcome this limitation, new approaches [7, 19, 9, 15] have been proposed where permissions are assigned to the users completely based on context information characterizing requestor's situations at access time. However, context-based access policies supported by these solutions are generally limited to context information characterizing requestor's situations. For example, they do not support policies based on owner's and resource's context information as follows: (i) *example of resource's context-based access policy:* a user grants access of read on her photos to nearby friends at photo shoot time; (ii) *example of owner's and resource's context-based access policy:* I grant access of read on documents created in my desk room to the team when I am located in company building.

We are mainly motivated by need to extend context-based access control models for expanding supported context dimensions that can be used for defining access policies completely based on context information. According to our knowledge, none of existing work take into account requestor's, owner's, and resource's context together for making access control decisions. Therefore, in this paper we propose an access control model that generalize the use of context information, offering to resource owners (or administrators) the possibility for defining access control policies completely based on owner's, requestor's, and resource's context.

The rest of paper is organized as follows: Section 2 presents our context model used by the proposed access control model. Then, we explain the proposed model in Section 3. Section 4 discusses the related work and, in Section 5, we present conclusion and future work.

2. MODELING CONTEXT

It is necessary to define a formal context model in order to facilitate context representation, sharing, and semantic interoperability in an access control framework implementing the proposed model. With this purpose, we defined an OWL DL ontology[1] for modeling context information that could be used for defining context-based access control policies. Our experience shows that using ontologies for context modeling is well suited for pervasive applications and services.

Before to present the proposed context model, named *Access Context Ontology*, let us define the concepts of `owner's context`, `requestor's context`, and `resource's context`, which describe the situation of *access entities*[2]. We have defined these concepts based on Dey's definition [4], as following:

- **Owner's context:** any information that can be used for characterizing the owner's situation of resources protected by the access control framework, which is relevant for making context-based access control decisions, such as location, activity, body temperature, blood pressure, etc. For example, in a context-based access control framework for Healthcare applications, a patient would like to grant permission of read on her medical records to any doctor if she is in a life-threatening situation characterized by a sudden drop in blood pressure or in heart rate. We could see in this example that the permission will be assigned to the requestors (i.e. any doctor) completely based on owner's context (blood pressure and heart rate);

- **Requestor's context:** any information that can be used for characterizing the entity's situation which is trying to access resources protected by the access control framework. For each access request received, the requestor's context is identified by the access control framework and it is used in order to determine access policies affected. For example, a user grants access of read on presentation file for everyone located in meeting room during the reunion (10-12h, November 3, 2009). In this case, the requestor's context (i.e. location and request time) is essential for taking access control decisions;

- **Resource's context:** any information that can be used for characterizing the situation in which the protected object was created and its current status, which is relevant for making access control decisions. For example, a user grants access of read on their photos to her friends that were located nearby her at photo shoot time. In this case, at the moment of a photo shoot (i.e. the creation time of resource) the resource is annotated with information describing the situation of creation (e.g. Bluetooth address of nearby mobile devices at the photo shoot time, in which will be used in order to infer the user's nearby friends).

Moreover, we are classifying the context information related with each *access entity* according to five dimensions considered by us as important for making context-based access control decisions: *spatial* - any information characterizing the situation from spatial dimension (e.g. location, place, GPS coordinates); *temporal* - any information characterizing the situation from time dimension (e.g. timestamp, period of day, month, year, day, season); *spatio-temporal* - any information characterizing the situation that is dependent of both spatial and temporal dimensions i.e. each piece of information is associated with a particular location at a particular time (e.g. weather conditions, temperature, noise,

[1]http://www.w3.org/TR/owl-guide/

[2]We use the term *Access Entity* to refer any implicated element of an access control system: owner, requestor, and resource.

luminosity); *social* - any information characterizing the situation from social relationships (e.g. nearby persons and nearby friends[3]); and *computational* - any information describing the situation from the computational characteristics (e.g. user's device capacities).

From our point of view, context information characterizing the environment (e.g. luminosity, temperature, noise) could be used for making access control decisions if and only if it is describing the context of an *access entity*. Normally this information is dependent of location and time dimensions, e.g. the value of temperature is valid if and only if this information is associated with a specific location at a particular time. We classified this information in our ontology as concepts describing the spatio-temporal dimension.

We have proposed in previous works the *Context top Ontology*[17, 16] for modeling the context dimensions listed above. We are reusing this ontology as basis to define our *Access Context Ontology*[4] (see Figure 1), which was extended in order to describe the context of any *access entity* supported by our access control model. Originally, this ontology was defined for modeling the annotation context of multimedia resources (see the PhotoMap application in [17, 16]). Moreover, we are reusing GeoRSS[5] concepts to describe GPS coordinates and spatial geometric relations, OWL-Time[6] to express temporal content, and RDF FOAF[7] ontology for describing social context dimension. From the *Context* concept described in the *Context Top Ontology*, we defined a subclass named *Access Context* (i.e. $AccessContext \subseteq Context$). This concept capture from the context any information characterizing the *access entities*, in which is relevant for making context-based access control decisions, i.e. it can be used for defining context constraints on access control policies. The format of context-based access control policies supported by the proposed model will be described in more detail in Section 3.

Figure 1 shows the proposed *Access Context Ontology*, which defines the principal context concepts for making context-based access control decisions that are describing the situation of *access entities*: Identity, Location (Indoor and/or Outdoor), FOAF profiles, Activity, Instant (time), period of day, etc. User's Identity can be semantically described by userID, user name, pseudonym, and role/group (i.e. when is possible to identify it), which are represented as datatype properties of *Identity* concept. We are using IETF RFC 4119[8] for describing semantically indoor and outdoor locations of users and geo-located concepts. The indoor location can be described using the following formats: building name (LMK); building name and floor (LMK, FLR); building name, floor, and room (LMK, FLR, LOC). The outdoor location can be completely stated using 4 (four) standard notations: (country); country and city (country, A3); country, city, and street (country, A3, A6-STS); country, city, street, and house number with suffix (country, A3, A6-STS, HNO-HNS).

The FOAF ontology was extended in order to define new subclass for classifying the social relationships existing between owners and requestors (e.g. family member, friend, bestFriend, supervisor, team member, etc). These social relationships will be explored by owners in the administration task of access control policy definition. This process will be described in more detail in Section 3.

[3]We means by nearby persons or nearby friends the social relationships associated to the situation that the system could infer from Bluetooth addresses of nearby user's mobile devices

[4]http://membres-liglab.imag.fr/bringel/AccessContext.owl

[5]http://www.georss.org/

[6]http://www.w3.org/TR/owl-time

[7]Friend of a Friend Ontology (http://xmlns.com/foaf/spec/)

[8]A Presence-based GEOPRIV Location Object Format. (http://www.ietf.org/rfc/rfc4119.txt?number=4119)

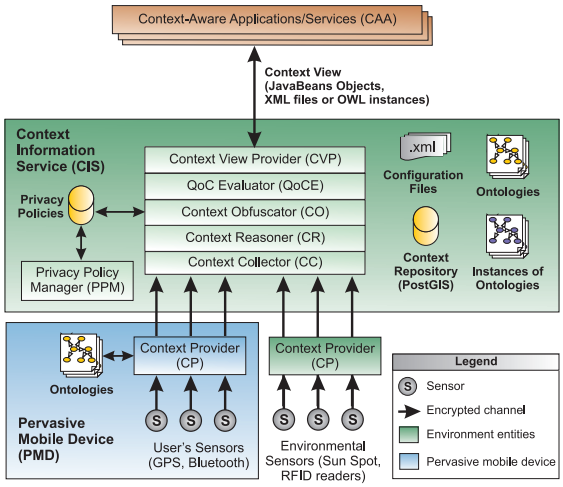


Figure 2: Our context management framework.

2.1 Managing Access Context Information

In order to deploy a context-based access control framework implementing our access control model, we should have a context management service providing context information. The context information will be verified by the access control framework for making context-based access control decisions. Figure 2 shows our context management framework, which was implemented[9] in order to support context-aware applications/services and context-based access control frameworks developed by our team, such as PhotoMap [17], QACBAC [6], and SW3A [18]. The key modules of our framework are:

- **Context Provider (CP):** CP are brokers present on environment (e.g. a server controlling sensors distributed in a building) or on mobile devices (e.g. user's sensor-rich mobile devices) sending captured context information to the *Context Information Service (CIS)* (more specifically to the *Context Collector - CC*). Each CP keeps a dynamic list of registered sensors (e.g. GPS, Bluetooth) controlling synchronous and asynchronous notifications from them. The collected context information is semantically represented using the *Access Context Ontology*;

- **Context Information Service (CIS):** this entity is the principal software component of our context management framework, which is composed by following sub-services:

[9]Java technologies: we are using J2EE for implementing the server side and J2ME to the client side.

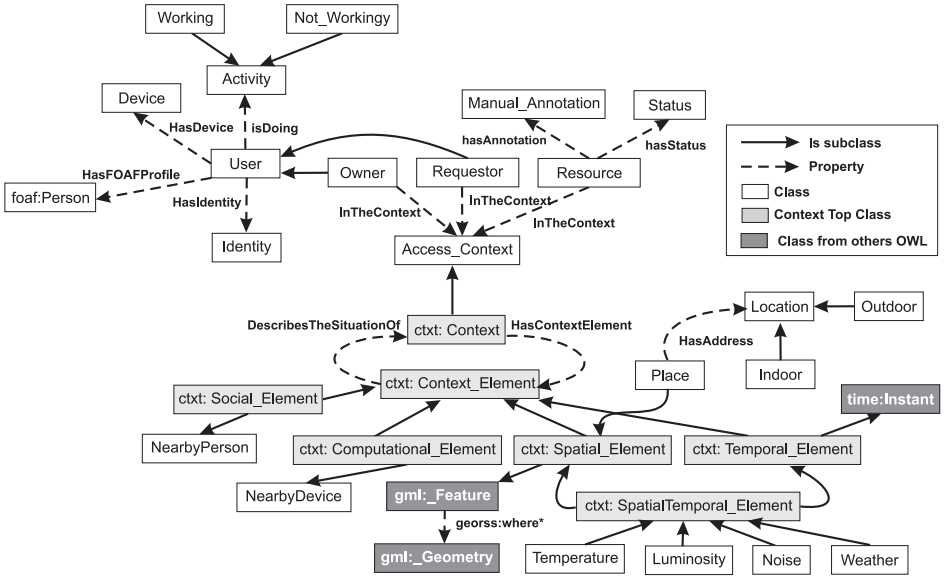


Figure 1: Access Context Ontology.

i) *Context Collector (CC)*: this service collects and aggregates context information sent by CP, storing as OWL documents that are instances of the Access Context Ontology. Moreover, context information described in these documents is stored by the Global Context Repository (GCR), which is a postGreSQL database server with the PostGIS[10] extensions;

ii) *Context Reasoner (CR)*: CR runs inference and derivation process on context information described by OWL documents in order to obtain semantic high-level context information i.e. enriching the user context. For instance, it is able to infer the name of nearby user's friends using the captured Bluetooth addresses of nearby devices and the user's FOAF profiles. This service requests some Web Services (e.g. georeverse Web Service from Geonames[11]) in order to have high-level semantic information from low-level semantic information. For example, it is able to derive the real address (i.e. place concept) from the GPS coordinates;

iii) *Context Obfuscator (CO):* this service enforces user's privacy policies on context information by running obfuscation and anonymization process based on ontologies. Privacy policies are represented on our system using SWRL[12] (Semantic Web Rule Language) that is a proposal W3C for a Semantic Web rules language. From the instances of *Access Context Ontology*, only the context concepts, relations, and datatype properties disclosed by the owner's context at request time will be described on the OWL document that will be sent to context consumers. We named this sub-set of context information described by this OWL document as *Context View* (i.e. $ContextView \subseteq AccessContext$). The privacy policies are stored by the*Privacy Policy Manager (PPM)*, which provides to the users a web interface for writing easily SWRL rules;

iv) *QoC Evaluator (QoCE):* it is the principal service for evaluating the Quality of Context [3]. It uses *Context Views* originated by the CO with QoC information captured from the environment (e.g. currentTime) and configuration files (e.g. lifeTime) in order to measure the QoC indicators associated with each context concept described by the *Context View* instance (e.g. the precision of location, up-to-dateness of temperature);

v) *Context View Provider (CVP):* context information queries are answered by the CVP. He is providing user's *Context Views* describing context information to the context consumers (e.g. the access control framework) as JavaBeans objects, XML files or OWL instances. CIS can yet to receive/transfer user's context information to other CIS belonging to different domains with which maintain trust relationships.

Moreover, the communication channels established between S, CP, CIS, and CAA are protected using protocols and security services defined on application layer. We are reusing the security framework [5] defined by us to construct context-aware security services. This requirement is essential to save the confidentiality, integrity, and authenticity of *Context Views*, from the capture to its use by context-aware application and services.

3. A GENERALIZED CONTEXT-BASED ACCESS CONTROL MODEL

The proposed access control model is centered on the context concept, which is composed by information characterizing the controlled resources, requestors, owners, and the environment surrounding them. In this model, the context act as a mediator between the entities requiring access to resources (i.e. requestors) and the set of permissions assigned to these resources. Context-based access control policies defined by resource owners (or an administrator) describe the context conditions, named `Access Control`, which must be met in order to obtain access to associated resources, i.e., requestor can perform only those operations related with the

[10]http://postgis.refractions.net/
[11]http://www.geonames.org/
[12]http://www.w3.org/Submission/SWRL/

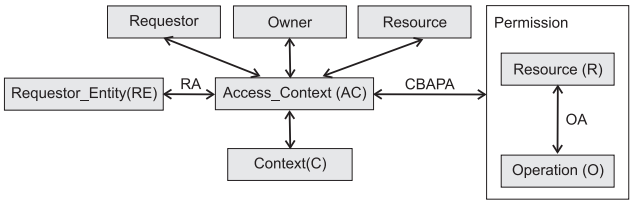


Figure 3: Generalized Context-Based Access Control Model.

access contexts currently active.

The specification of our model is based on RBAC model but we are using the *Access Context* concept defined in the *Access Context Ontology*, which represent context information sentences relevant for taking access control decisions, instead of roles as being the main difference. Therefore, the permissions are assigned to requestor entities taking into account *Access Context* assigned to context-based access policies, which are defined by resource owners or by administrator of access control system.

Access policies are defined describing relationships between 6 (six) set of elements of our access control model: *Requestor Entities(RE), Context(C), Access Context(AC), Resources(R), Operations(O), and Permissions(P).* Figure 2 shows our access control model and the relationships between these elements.

Definition 1. The generalized Context-Based Access Control Model is composed of the following components:

- A set RE of requestor entities, a set AC of access context, a set R of resources, a set O of operations, and a context condition language *CCL*.
- A set of permission P ($P = 2^{R \times O}$) $= \{(r, o) | r \in R, o \in O\}$, which each permission is a approval to perform an operation on one resource in a given *Access Context*.
- The set of *context-based access permission* CBAP ($CBAP = 2^{AC \times P}$) $= \{(ac, p, e) |\ ac \in AC,\ p \in P,\ e\ is\ a$ expression of *CCL* defined using the values of context concept in the ac$\}$.
- **Requestor entity Assignment (RA):** $RA \subseteq RE \times AC$, it is a many-to-many relationship mapping RE-to-AC assignment relation.
- **Context-based access permission Assignment (CBAPA):** $CBAPA \subseteq AC \times CBAP$. Each AC could be associated with many CBAP and each CBAP has only one AC assigned. It is a one-to-many relationship mapping AC-to-CBAP assignment relation.
- **Operation Assignment (OA):** $OA \subseteq O \times R$. Each operation (O) could be associated many resources, and for each resource (R) could be granted many operations. It is is a many-to-many relationship mapping operation-to-resource assignment relation.
- **Assigned requestor entities to access context: assigned_re(re, ac)** $= \{re \in RE, ac \in AC | (re, ac) \in RA\}$, the mapping of an *access context* onto a set of requestor entity.
- **Assigned permissions to access context: assigned_p(p, ac)** $= \{p \in P, ac \in AC | (p, ac) \in PA\}$, the mapping of *access context* onto a set of permissions.
- **Operations associated with permission:** $(p : P) \rightarrow o \subseteq O$, the permission-to-operation mapping. It is the set of operations associated with permission p.
- **Resources associated with permissions:** $(p : P) \rightarrow r \subseteq R$, the permission-to-resource mapping. It is the set of resources associated with permission p.
- In this access control model, there is an inheritance relationship between AC. It is described formally as below:
  $IAC \subseteq AC \times AC$ is an inheritance relation $(\Rightarrow) ac1 \geq ac2$, which means that *ac1* inherits the privileges of *ac2*.
  $\rightarrow assigned\_p(p, ac2) \subseteq assigned\_p(p, ac1)$

In what follows we provide additional details on the context condition language defined in order to describe condition expressions based on *Access Context* concepts defined by the *Access Context Ontology*.

3.1 Context Condition Language (CCL)

The proposed model includes a simple language for expressing constraints based on *access context* information. Such context constraints defined by means of expressions should be evaluated when enforcing context-based access control policies. The context constraints that could be expressed by CCL are defined as following:

Definition 2. Let c be a concept of AC *(Access Control)*. Each $c \in AC$ has a domain of possible values, denoted as *Dc*. An atomic context condition (acc) defined over AC has the form (c op v), where $c \in AC, v \in Dc$, $op \in OP = \{>, \geq, <, \leq, \neq, =\}$. The set of *op* can be extended in order to accommodate user-defined operators as well. The constraints of CCL are defined as following:

- An atomic context condition (acc) is a constraint of *CCL*;
- Let acc_i *and* acc_j be constraints of *CLL*, then $acc_i \wedge acc_j$ is a constraint of *CCL*;
- Let acc_i *and* acc_j be constraints of *CLL*, then $acc_i \vee acc_j$ is a constraint of *CCL*.

Based on this language, we are able for specifying any complex context constraint in order to describe all kinds of context-based access control policies supported by the proposed model.

3.2 Policy representation

Policies are used to mediate context-based access control decisions, which are described as tuples defined by owner's resources or by access control administrators. They represent associations between *Requestor Entity*, *Access Context*, and *Permission*. They are defined following the format below:

$$policy(pol) = (re, p, e, enable\_bit) \quad (1)$$

- *re* is a identity assigned to the requestor entity (e.g. name, group, role). When omitted or assigned the value *everyone*, only requestors associated to the access context *ac* that meets the constraint defined by *e* get access to the protected resource;

- *p* is a permission ($p \in P$);

- *e* is a context constraint expression defined using *CCL*;

- *enable_bit* indicates whether the associated policy is enabled/disabled (1,0).

Owner's resources could define also a set of policy that is represented formally as follows:

$$policy\_set(pol_{set}) = \{pi \mid pi \; is \; a \; policy, i \in [1, I]\} \qquad (2)$$

3.3 Enforcing request of access

We define a request of access (Req_A) as a triple *(req, perm, ac)*, where:

- *req* is a requestor entity who issues this data access;

- *perm* is the permission that this requestor wants to acquire.

- *ac* is a set of values for every context concept *c* described by *Access Context* at request time. That is, $ac = \{v_1 \; of \; c_1, \; v_2 \; of \; c_2, ..., \; v_n \; of \; c_n\}$, where $\{c_1, \; c_2, ..., \; c_n\}$ is the set of context concept described by the *Access Context*.

A request of access $Req_A(req, perm, ac)$ is granted only if there exists an access control policy $pol = (re, p, e, enable\_bit)$, such that $enable\_bit = 1$ (i.e. the policy is enabled), $req \in re$, $perm = p$, and e evaluates to true under ac (i.e. when all c_1 in constraint expression e are replaced with their values in ac, then the resulted Boolean expression is true).

From this definition, we can design algorithms in order to determine whether a request of access is authorized or not based on context values described by *Access Context*. We have proposed a solution divided into two algorithms: the first algorithm (see in Figure 4 algorithm 1) is in charge of identify the candidate policies (Pol_{setC}) from the policy set defined by the resource owners, which are been controlled by the *Policy Administration Point* of an access control framework in which implement the proposed model (we will describe in detail this entity in subsection 3.5); the second algorithm (see Figure 5) evaluates the context constraint expression associated with each candidate policy in order to identify if this expression is true for the current *access context*.

Algorithm 1 verifies for each Pol_i in the Pol_{setC} if the *req* of Req_A (i.e. request of access) is in the *re* of Pol_i. In additional, it is verified if the *perm* of Req_A is equal to *p* of Pol_i. If these two conditions are true, the Pol_i is a candidate policy. After running the algorithm for identifying the candidate policy set (Pol_{setC}), this set will be evaluated by the Algorithm 2 (EvaluateContextConstraint) in order to verify if the expression *e* is true for the current *access context*, grating/denying access permission to the requestor entity.

Algorithm 1: EnforceRequest(Req_A)

Require: Req_A
Ensure: Permit/Deny
Pol_{setC} ← Null *{initializing the set of candidate policies}*
Result ← Deny *{initializing the result}*
{Identifing the set of candidate policies}
for each $Pol_i \in Pol_{set}$
  if (enable_bit of Pol_i is true)
    if (req of $Req_A \in$ re of Pol_i) **and**
      (perm of Req_A= p of Pol_i)
      $Pol_{setC} \leftarrow Pol_{setC} + Pol_i$
    end if
  end if
end for

{Evaluating context constraints of candidate policies}
for each Pol_j in Pol_{setC}
  if (EvaluateContextConstraints(e of Pol_i) is true)
    result ← Permit
    Break
  else
    result ← Deny
  end if
end for
{Granting/denying access}
return result

Figure 4: Enforcing request of access.

3.4 Policy Examples

Using the concepts of *Access Context Ontology* for characterizing the *access entities* defined in Section 2, we can define 7 (seven) different sets of context-based access policies. In this section we present example of each set type, showing the expressiveness in which we could define context-base access policies using the *CCL*.

- **Owner's context-based access policy:** this type of policy is taking into account only context information characterizing the owner's context.

  Example: a patient (owner) grants permission of read on her medical records to any doctor if she is in a life-threatening situation characterized by a sudden drop in blood pressure (*blood_p*) or in heart rate (*heart_r*).

  $ac1 = \{(owner.blood\_p < 85) \vee (owner.heart\_r < 60)\}$;

  $Pol_1 : (doctor, (medicalrecords, read), ac1, true)$;

- **Resource's context-based access policy:** this policy type is taking into account only context information characterizing the resource's context.

  Example: a user grants access of read on photo_collection1 to everyone nearby her mobile device at photo shoot time.

  $ac2 = \{(requestor.deviceAddr \; in \; resource.nearbyDeviceAddrs)\}$;

  $Pol_2 : (everyone, (photo\_collection1, \; read), ac2, true)$;

Algorithm 2: EvaluateContextConstraint(e)

Require: e, ac
Ensure: true/false
result ← false
{Verifing each access context condition}
for each $acc_i \in e$
  v ← value of c_i described in ac
  if (acc_i is false with the value v)
    return false
    break
  end if
end for
return true

Figure 5: Evaluating context constraint.

- **Requestor's context-based access policy:** this policy group is taking into account only context information characterizing requestor's context.

  Example: a user grants access of read on $presentation\_file$ to everyone located in the meeting room.

  $ac3 = \{(requestor.location = \text{“}meeting\_room\text{”})\};$

  $Pol_3 : (everyone, (presentation\_file, read), ac3, true);$

The following policy examples are defined using context information characterizing simultaneously one or more *access entities*.

- **Owner's and resource's context-based access policy:** when I am located in my desk room (owner's context), I grant access of read on the documents created in this room (resource's context) to my team.

  $ac4 = \{(owner.location = \text{“}desk\_room\text{”}) \wedge (resource.location = \text{“}desk\_room\text{”})\};$

  $Pol_4 : (team, (resource, read), ac4, true);$

- **Owner's and requestor's context-based access policy:** I grant access of read on $photo\_collection2$ to everyone located nearby (owner's and requestor's context).

  $ac5 = \{(requestor.deviceAddr\ in\ owner.nearbyDeviceAddrs)\};$

  $Pol_5 : (everyone, (photo\_collection2, read), ac5, true);$

- **Requestor's and Resource's context-based access policy:** a user grants access of read on photos taken in Paris (resource's context) to everyone located in this city (requestor's context).

  $ac6 = \{(requestor.location = \text{“}Paris\text{”}) \wedge (requestor.location = photo.location)\};$

  $Pol_6 : (everyone, (photo, read), ac6, true);$

- **Generalized context-based access policy:** a user grants access of write photo taken in Paris (resource's context) to everyone located in this city (requestor's context), but only when she is located also in Paris (owner's context).

  $ac7 = \{(requestor.location = \text{“}Paris\text{”}) \wedge (requestor.location = photo.location) \wedge (requestor.location = owner.location)\};$

  $Pol_7 : (everyone, (photo, write), ac7, true).$

In this example (Pol_7), we can identify yet a inheritance relationship between $ac7$ (Pol_7) and $ac6$ (Pol_6), which means that each requestor associated to $ac7$ (this context is more restrictive that $ac6$, and $ac6$ is included in $ac7$) inherits the privileges assigned to $ac6$ (i.e. (photo, read)).

3.5 Implementation

In order to deploy an access control framework supporting each type of context information in which can affect context-based access control decisions, it is necessary that the following requirements are met:

- Support for sensing and management of context information associated with requestors and owners, which should be available to the access control framework for making access control decisions;
- Support for sensing context information at creation time of resources, providing mechanisms to assist the users in the task of context annotation associated with the resources;
- It is required to guarantee the security, privacy, and quality of context information used for making context-based access control decisions, as we identified in [6]. Compromised context information could cause incorrect access control decisions generating security breaches in the access control system;
- In addition to determining context information at request time of resources and deciding whether to permit access, it should be possible to suspend a permission assigned to the current context when it changes to a state where the context constraint expression is not more true [19].

We have identified two implementation approaches for assigning permissions based on context: the passive and the active approach. The first approach (passive) is characterized by the context verification only at request time of resource, identifying the affected permissions and granting/denying access to requestor, i.e. the access control framework identifies the owner's, requestor's, and resource's context at request time and enforces the affected access control policies. The second approach is more complex, which it is necessary constantly verifying context information in order to identify assigned permissions described in the policies. When an active permission is identified, the access control framework notifies the affected users (e.g. the access control interface shows the resource names when it is active for the current context). The task of assigning permission is dynamically context-dependent.

The passive approach is the most implemented solution by context-based/aware access control systems, primarily motivated by simplicity of implementation. It requires the disclosure of the list of resources controlled by the access control framework in order to allow requestors to query access for any listed resources. This approach is lightweight but presents a security breach resulting of the disclosure of

the list of resources controlled by the access control system, even if they do not have permissions to access the resources.

The active approach requires a notification service that constantly updates the list of resources available to the affected users, taking into account the permissions assigned to the context conditions (i.e. access policies) that match the current context.

An intermediate approach is still possible to be implemented, where the process is conducted in two phases. Firstly, the user requests a list of resources active taking into account the current context. Posteriorly, she requests access to the resources from this list, in which the current context will be verified again in order to validate the permissions for the requestor's context. This approach eliminates the security problem present in the passive approaches, beyond reducing the consumption of resources and the difficulty for implementing the active approach.

We are developing our access control model based on the intermediate approach in order to assign context-based permissions. Moreover, we are taking into account the requirements identified above. The first three requirements are directly met through the Context Management Architecture presented in Section 2. The last requirement is met implementing the algorithm 2 described in Section 3 in order to evaluate each permission assigned to the users when occurs changes on context.

Our access control framework is based on Sun's XACML Implementation[13], which we are considering that all context-based access control decisions will be performed on server side. The idea is to validate our approach in a client/server scenario and afterward propose an access control light-middleware that will be executed completely on mobile devices (i.e. requesting and enforcing context-based access control decisions on resources). XACML is an OASIS[14] standard that describes both a policy language and an access control decision request/response language (both encoded in XML).

XACML defines a policy language using attributes for describing requestors, resources, and environment. From our point of view, these XACML attributes have the same function in the XACML access control decision process that context information have on our access control approach. Thus, we extended the Sun's XACML implementation in order to support context information describing the situation of the *access entities* as attributes.

Figure 6 shows the proposed access control framework in which is integrated with our context management architecture. The access control framework mainly contains PEP (Policy Enforcement Point), PDP (Policy Decision Point), PCP (Policy Context Information Point), PAP (Policy Administration Point). The PCP is a software entity based in the PIP entity (Policy Information Point) of the Sun's XACML implementation, which was modified in order to support context information as attributes. Originally, this entity is in charge of transfer the attributes required for making ACML access control decisions.

The flow of an request of access is as follows (see Figure 6): (1) the entity (e.g. a user, a service) that tries to access a resource sends a request described using the XACML request language to the PEP; (2) the PEP intercepts this request and sends the requests to the PDP; (3-6) the PDP makes access decisions according to the access policy or policy set written and managed by PAP, using the context attributes obtained by querying the PCP (i.e. it request the access context to the CVP of our context management architecture); (7) the access decision given by the PDP is sent to the PEP and the PEP permits or denies the access request according to the decision of PDP (8).

Moreover, the PDP was modified in order to filter the policy set using global requirements of QoC (QoCR) defined by the applications and/or users. If an access policy does not attain these QoCR, it will be out of the policy set that will be enforced by PDP. As result, we improve the performance of our access control framework.

The response always includes an answer about whether the request should be allowed using one of four values: Permit, Deny, Indeterminate (an error occurred or some required value was missing) or Not Applicable (the request can't be answered).

The PEP and PDP might both be contained within a single application (e.g. on mobile devices), or might be distributed across several servers. In the current version of our access control framework, the user devices have only the PEP deployed and request the PDP contained in a server for making access control decisions. We offer a graphic tool (Web interface), called Policy Creator, in order to help users to build their own XACML policy documents in a friendly and easy way.

4. RELATED WORK

We have identified and classified access control approaches that use context information for making access control decisions on two types: i) `context-aware` access control and ii) `context-based` access control. The first group is composed by approaches that use context information as a way for assigning dynamically roles/identity and associated permissions to the users, i.e., context information is only used for improving an existing model that is not context-dependent. On the one hand, they could work without context information. On the other hand, the expressiveness of access control policies will be limited.

Some research [2, 11, 13, 20, 10] attempts have been done under the first group, which take into account context information as an optional attribute used for limiting the scope of access control policies, i.e. they are aware of context. Bertino et al. have proposed in [2] the Temporal Role-Based Access Control Model (TRBAC) to add up the time dimension and the concept of role enabling/disabling in order to improve the RBAC model. Ray et al. [13] extended RBAC towards a Spatial-Temporal Role-Based Access Control Model for taking into account both spatial and temporal context dimensions. Moyer et al. presented a Generalized Role-Based Access Control model (GRBAC) [11], which extend the context dimensions supported by incorporating the notion of object roles and environment roles into RBAC model. In [20] Zhang et al. proposed the Dynamic Access Control Model (DRBAC) in order to deal with context information and Kim et al. [10] proposed an similar approach that extend RBAC model for adjusting dynamically role assignments (UA) and permission assignments (PA).

Similarly as occurs with RBAC model, all these RBAC-based approaches do not work well in PCE where we cannot assume the existence of predefined roles and user-role associations, which the relationships between resource requestors

[13]http://sunxacml.sourceforge.net/

[14]http://www.oasis-open.org/home/index.php

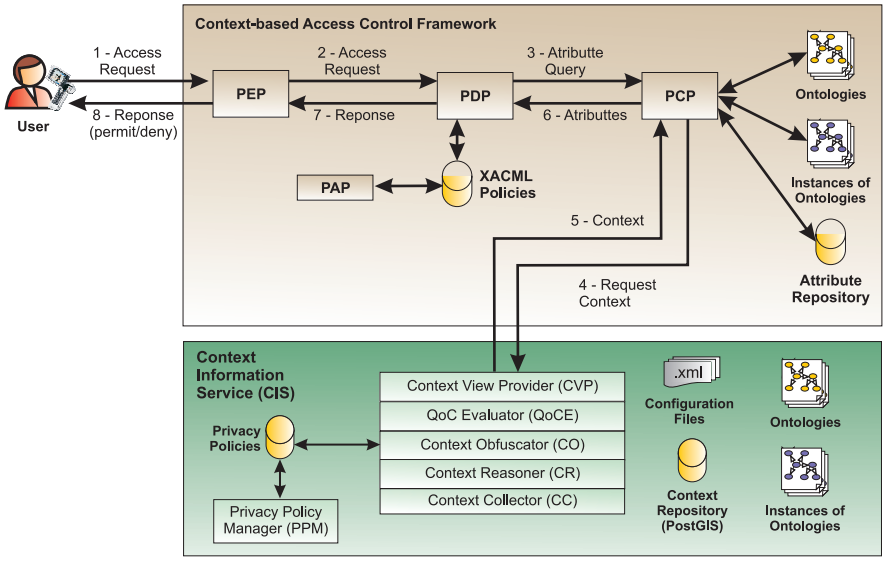


Figure 6: The proposed Access control Framework.

and resource owners tend to be ad-hoc in nature. For example, the complete list of users on the environment may not be known, making it infeasible to define access control policies based on role-user associations.

The second group is composed by context-centric solutions where context is the principal entity explicitly guiding both specification and enforcement of access control policies. Instead of assigning permissions directly to the users and defining context-aware restrictions in which these permissions should be applicable, resource owners define for each resource the context conditions that enable someone to access it, i.e., the access control policies are completely based and dependent of context information. When a request on protected resource is done the access control framework identifies the current context in order to enforce associated access control policies, grating/denying access to the requestor.

Some research [19, 9, 15] have followed this direction, which permissions are assigned to the users based on the current situation. Yokoyama *et al.* [19] have proposed an Anonymous Context Aware Access Control Architecture (ACA2) based on an analogy to the public telephone service, where users can anonymously access services supported by their context through pre-registered software components (proxies). The main features of this architecture include anonymity, access suspension caused by context changes, and active context certificates with stream verification. Since this solution is focusing on architecture, it is not so clear the formalization of the proposed access control model. Groba *et al.* [9] have presented a context-dependent access control for context information, which can be characterized by three basic properties: owner-centric, context-dependent, and individual role model for each user. However, this solution has been proposed to control access exclusively on context information, which limits its use in PCE (resources can be data, services, etc).

To the best of our knowledge, the work more closer to our proposition is described in [15], where the authors present a semantic access control aproach based on a context-aware policy model, which treats context as a first-class principle for policy specification and adopts a hybrid approach to policy definition based on DL ontologies and LP rules. However, as occurs with the existing approaches from the second group, this approach present a limited support to the context dimensions (e.g. resource's context), mainly based on requestor's context and the collocation concept, meaning that the resource owner is located on the same place in which the resource requestor is (i.e. she is collocated with the requestor).

Moreover, all these propositions from the second group do not present a formal definition of the access control model used in order to implement their solutions. In this paper, we define clearly and formally the proposed context-based access control model, presenting the context dimensions associated to the owners, requestors, and resources, which could be used for making context-based access control decisions. This generalized context-based access control model could be utilized in order to implement and to combine various kinds of access control concepts using context information, which could be tailored and extended in order to reduce/enlarge the support for context-based access control decisions. For example, the role concept is supported by our access control ontology, permitting to extend the proposed model to a hybrid approach based on both context and roles (i.e. when it is possible to determine and assigning roles to the users).

5. CONCLUSIONS

Open, dynamic, and heterogeneous pervasive environments call for new access control solutions, requiring changes in the focus of access control models from identity/role-based to the context-based approaches.

With this objective in mind, in this paper we propose a generalized context-based access control model that offer to resource owners and access control administrators the possibility for defining context-based access policies taking into account context information describing the owner's, re-

questor's, and resource's situations. According to our knowledge, none of the existing work considers the owner's, requestor's, and resource's context together for making context-based access control decisions in PCE.

Therefore, the proposed model extends the support for defining access policies completely based on context information, offering 7 (seven) types of context-based access control policies.

We are currently working on implementing a prototype integrating this model with the QACBAC model proposed by us in [6], in order to enforce access control policies taking into account both: the generalized use of context information and associated QoC requirements. We have identified in [6] that using context information with inadequate QoC may increase the probability of incorrect access control decisions. We are integrating a conflict detection mechanism for

Moreover, we plan to extend the proposed model in order to take into account the privacy requirements when enforcing access control policies, such as the support to purposes and obligations. In additional, we plan to integrate a mechanism to dynamically and statically detect/resolve conflict on context-based access control policies.

The focus of our work has been predominantly on context-based access control, then our idea is to propose a family of context-based access control models for pervasive environments that can be implemented taking into account specific application/service requirements, such as policy hierarchies, context dimensions, quality of context, and privacy.

6. REFERENCES

[1] Trusted computer system evaluation criteria, dod 5200.28-std, department of defense, 1985.

[2] E. Bertino, P. A. Bonatti, and E. Ferrari. Trbac: a temporal role-based access control model. In *ACM Workshop on Role-Based Access Control*, pages 21–30, 2000.

[3] T. Buchholz, A. Küpper, and M. Schiffers. Quality of context: What it is and why we need it. In *(HPOVUA 2003), Geneva, 2003*, 2003.

[4] A. K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

[5] B. Filho, W. Viana, R. Braga, and R. Andrade. Framesec: A framework for the application development with end-to-end security provision in the mobile computing environment. In *AICT-SAPIR-ELETE '05*, pages 72–77, Washington, DC, USA, 2005. IEEE Computer Society.

[6] J. B. Filho and H. Martin. Qacbac: an owner-centric qoc-aware context-based access control model for pervasive environments. In *SPRINGL '08: Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pages 30–38, New York, NY, USA, 2008. ACM.

[7] J. B. Filho and H. Martin. A quality-aware context-based access control model for ubiquitous applications. In *ICDIM, Third IEEE International Conference on Digital Information Management (ICDIM), November 13-16, 2008, London, UK, Proceedings*, pages 113–118, 2008.

[8] J. B. Filho and H. Martin. Using context quality indicators for improving context-based access control in pervasive environments. *Embedded and Ubiquitous Computing, IEEE/IFIP International Conference on*, 2:285–290, 2008.

[9] C. Groba, S. Grob, and T. Springer. Context-dependent access control for contextual information. In *ARES '07: Proceedings of the The Second International Conference on Availability, Reliability and Security*, pages 155–161, Washington, DC, USA, 2007. IEEE Computer Society.

[10] Y.-G. Kim, C.-J. Moon, D. Jeong, J.-O. Lee, C.-Y. Song, and D.-K. Baik. Context-aware access control mechanism for ubiquitous applications. In *AWIC*, volume 3528 of *Lecture Notes in Computer Science*, pages 236–242. Springer, 2005.

[11] M. J. Moyer and M. Ahamad. Generalized role-based access control. In *ICDCS*, pages 391–398, 2001.

[12] S.-H. Park, Y.-J. Han, and T.-M. Chung. Context-role based access control for context-aware application. In *High Performance Computing and Communications, Second International Conference, HPCC 2006, Munich, Germany, September 13-15, 2006, Proceedings*, volume 4208 of *Lecture Notes in Computer Science*, pages 572–580. Springer, 2006.

[13] I. Ray and M. Toahchoodee. A spatio-temporal role-based access control model. In *Data and Applications Security XXI, 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Redondo Beach, CA, USA, July 8-11, 2007, Proceedings*, volume 4602 of *Lecture Notes in Computer Science*, pages 211–226. Springer, 2007.

[14] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

[15] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila. A semantic context-aware access control framework for secure collaborations in pervasive computing environments. In *International Semantic Web Conference*, pages 473–486, 2006.

[16] W. Viana, J. B. Filho, J. Gensel, M. Villanova-Oliver, and H. Martin. Photomap - automatic spatiotemporal annotation for mobile photos. In *Web and Wireless Geographical Information Systems, 7th International Symposium, W2GIS 2007, Cardiff, UK, November 28-29, 2007. Proceedings*, pages 187–201, 2007.

[17] W. Viana, J. B. Filho, J. Gensel, M. Villanova-Oliver, and H. Martin. A semantic approach and a web tool for contextual annotation of photos using camera phones. In *WISE*, pages 225–236, 2007.

[18] W. Viana, J. B. Filho, J. C. F. Junior, G. J. de Sena, E. L. F. Senne, J. Gensel, M. Villanova-Oliver, and H. Martin. Caus: Uma arquitetura para sistemas de ensino. In *SBCUP*, 2009.

[19] S. Yokoyama, E. Kamioka, and S. Yamada. An anonymous context aware access control architecture for ubiquitous services. In *7th International Conference on Mobile Data Management (MDM2006), Nara, Japan, May 9-13, 2006*, page 74. IEEE Computer Society, 2006.

[20] G. Zhang and M. Parashar. Context-aware dynamic access control for pervasive computing, 2004.

Towards Location-Based Access Control in Healthcare Emergency Response

- Position paper -

Carmen Ruiz Vicente
Aalborg University
carmrui@cs.aau.dk

Michael Kirkpatrick
Purdue University
mkirkpat@cs.purdue.edu

Gabriel Ghinita
Purdue University
gghinita@cs.purdue.edu

Elisa Bertino
Purdue University
bertino@cs.purdue.edu

Christian S. Jensen
Aalborg University
csj@cs.aau.dk

ABSTRACT

Recent advances in positioning and tracking technologies have led to the emergence of novel location-based applications that allow participants to access information relevant to their spatio-temporal context. Traditional access control models, such as role-based access control (RBAC), are not sufficient to address the new challenges introduced by these location-based applications. Several recent research efforts have enhanced RBAC with spatio-temporal features. Nevertheless, the state-of-the-art does not deal with mobility of both subjects and objects and does not support the utilization of complex access control decisions based on spatio-temporal relationships among subjects and objects. Furthermore, such relationships change frequently in dynamic environments, requiring efficient mechanisms to monitor and re-evaluate access control decisions. In this position paper, we present a healthcare emergency response scenario which highlights the novel challenges that arise when enforcing access control in an environment with moving subjects and objects. To address a realistic application scenario, we consider movement on road networks, and we identify complex access control decisions relevant to such settings. We overview the main technical issues to be addressed, and we describe the architecture for policy decision and enforcement points.

Categories and Subject Descriptors

H.2.0 [**General**]: Security, integrity, and protection; H.2.8 [**Database Applications**]: Spatial databases and GIS; J.3 [**Life and Medical Sciences**]: Medical Information Systems

1. INTRODUCTION

The availability of mobile devices with positioning capabilities has fostered the development of location-based applications that allow users to access information relevant to their spatio-temporal context. For instance, visitors of a museum may be able to access a electronic on-line guide system as long as they are situated within the museum's perimeter. Such applications introduce specific security challenges that reach beyond the capabilities of traditional access control systems, such as Role Based Access Control (RBAC).

Several models have been proposed [1, 2, 5] that extend RBAC to allow the specification of policies in which the access decision is determined by the spatio-temporal context of subjects and objects. However, these models relate the role extents to fixed locations, restricting the policies to static zones. Therefore, they may not be suitable for dynamic mobile environments where both subjects and objects are moving continuously. In addition, in certain location-based application scenarios, the access control decision may depend on complex spatio-temporal relationships among subjects and objects. Surveys of access control in location-based services can be found in the literature [3].

In this position paper, we outline an access control framework for healthcare emergency response that illustrates the requirements to access control in a dynamic environment with mobile subjects and objects. In this scenario, a patient who requires an emergency response triggers an *event* that results in the creation of an emergency care request in the system. Such an event can be generated by the patients themselves, or by an automatic system designed to alert emergency response teams (e.g., in-vehicle GPS-enabled units such as On-Star provide automatic car crash notification).

The record inserted in the system contains a number of *event attributes*, for instance, symptoms, vital signs, as well as the age and gender of the patient. Based on the patient identifiers (e.g., cell phone number, or identifier of the reporting On-Star unit) additional information can be associated with the emergency record request, such as the patient's medical history.

The location of the emergency, the patient symptoms, medical history, etc., represent *objects* that need to be accessed by several categories of mobile *subjects*, such as medical doctors, ambulance personnel or medical-trained volunteers. Note that the objects may also be mobile. For instance, a patient suffering from a heart attack may be in a moving vehicle drove by a family member.

An event may refer to a single patient, e.g., an individual with a heart attack, or it may involve several patients, e.g., in the case of a car accident. In response to an event, the emergency response system must find subjects that are authorized to access the event's attributes and provide on-site help to patients. We identify two key requirements.

First, as medical information is sensitive, the system needs to ensure that access to confidential data is thoroughly controlled. Note that distinct subjects may have different privileges with respect to information disclosure: For example, doctors may be allowed to

access medical history, whereas volunteers should not learn confidential information about the patient that it is not related to the current event.

Second, an emergency requires fast response. The authorized subjects should be able to arrive at the location of the emergency within a maximum period of time. Thus, the decision about which subjects to authorize must take into account the distance between the event location and the subjects.

Two representative types of location constraints are *range* and *nearest-neighbor* constraints. For instance, the former corresponds to requirements such as, *"A subject is only allowed to access an event record if its distance to the event site is less than one mile,"* whereas the latter addresses requirements such as, *"A subject is only allowed to access an event record if it is the nearest subject to the event site."*

In addition, the positions of both subjects and objects may change over time. Therefore, the spatial relationship between subjects and objects must be monitored continuously. Note that spatial and non-spatial (e.g., level of medical education, specialization, required equipment) constraints may be combined. For instance, if a volunteer and a doctor are located at the same distance, only the doctor should be granted access. Such complex access control decisions require specialized mechanisms for the design of policy enforcement and decision points (PEP/PDP).

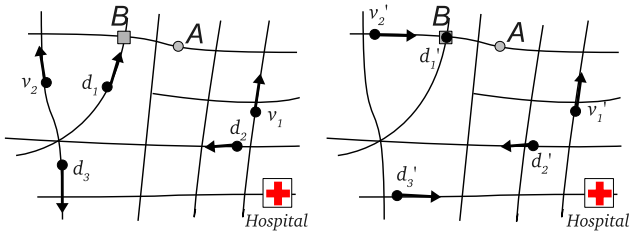


Figure 1: Emergency Response Scenario

The example in Figure 1(a) shows several volunteers v_i and off-duty doctors d_i driving around a city, in additio to a hospital. Consider that a car accident is reported at location A, and the car crash victim has a pre-existing heart condition and suffers from diabetes. In response to the event, the emergency response system takes the following actions: First, an ambulance is requested from the nearest hospital. Next, doctor d_1 who is nearest to the accident site is notified, and starts moving towards the site to perform a preliminary evaluation and treatment before the ambulance arrives. The notification sent to d_1 includes the coordinates of the accident site together with some basic information about the nature of the emergency, the victim's age, and pre-existing conditions.

As d_1 moves towards the accident site, a traffic jam may occur at intersection B, so d_1 is no longer able to reach the patient in time (Figure 1(b)). The system considers additional subjects, such as volunteers v_1 and v_2. Although v_2 is closer to the accident site than v_1, the route of v_2 crosses the traffic jam, so v_1 is chosen instead. In addition, doctor d_2 is selected, and both are granted access to the event record. However, doctor d_2 is granted permission to access more data about the victim, such as medical history. Such information is not made available to v_1.

While the volunteer v_1 and the doctor d_2 stabilize the patient, the hospital may not find an available ambulance. Then, the system may authorize d_2 to drive the patient to the hospital, if the patient is in a stable condition. Otherwise, an additional doctor (e.g., d_3) may be authorized to assist in the emergency. The system may decide on the amount of data to be disclosed to a subject based on the proximity to the accident site. This prevents the disclosure of sensitive information to subjects that are not likely to participate in the emergency, such as subjects that are delayed due to environmental conditions (as d_1 in the example above). However, some general details can be provided to such entities, in order to provide a backup plan.

This approach applies to location information as well: for instance, only coarse-grained location information may be disclosed to remote subjects, whereas accurate location data is sent only to nearby subjects.

The objective of this position paper is to identify the representative requirements and challenges of location-based access control in a dynamic environment with mobile subjects and objects. We investigate access control decisions and enforcement with respect to a combination of spatial and non-spatial (e.g., role-based) constraints.

Although our focus is on healthcare emergency response, many of the aspects discussed are relevant to other classes of location-based applications as well (e.g., public transportation). We emphasize that our work discusses initial directions and identifies key challenges in location-based access control; the realization of a complete access-control mechanism is beyond the scope of this paper. However, we do give an overview of the architectural components involved, as well as the main functionality offered by each component.

The rest of the paper is organized as follows. Section 2 describes the event and policy models. Section 3 illustrates the proposed architecture of the system and the tasks performed in each component. Section 4 outlines future research directions.

2. EVENT AND POLICY MODELS

As mentioned in Section 1, our envisioned access-control system is event-centric, meaning that objects are generated following the occurrence of an event such as a medical emergency (e.g., car accident). After object creation, the system must decide which subjects (e.g., medical doctors, volunteers) are granted permission to access the event-related data based on a mix of spatial and role-based constraints.

2.1 Event Model

An **event** is specified as a tuple:

$$E = \langle location, category, maxResponseTime, attributes \rangle$$

where

- $location$ indicates the position of the event
- $category$ describes the event type (different types require different numbers and types of entities)
- $maxResponseTime$ indicates the maximum allowed time within which the event must be handled (e.g., the maximum allowed time for the responders to arrive)
- $attributes$ is a list of event characteristics

Depending on the required functionality of the system, $location$ may be specified either in physical terms (e.g., as GPS coordinates) or logical terms (e.g., at the corner of Main Street and Elm Drive). The use of $category$ determines the number of responders and their roles. For example, the organization may stipulate that $category = carCrash$ requires two *doctors* and one *volunteer*.

Indicating a $maxResponseTime$ authorizes the system to relax the data confidentiality and other constraints if the request cannot be fulfilled. This mechanism allows flexibility for events that require immediate focus and response, such as severe automobile crashes. However, we also assume that most events will provide this piece of data to ensure that help is provided within a reasonable timeframe.

The $attributes$ are provided when an event is triggered (e.g., the cell phone number where the 911 call originated). Based on such attributes, additional data objects can be originated, such as the age and gender of the person who owns the cell phone (such data is relevant if the victim places the call).

2.2 Policy Model

The system must support both traditional (e.g., role-based) and spatio-temporal policies. Traditional policies can integrate other access control or RBAC rules, such as *"A volunteer is not allowed to access a patient's previous medical records."*

Spatio-temporal policies indicate constraints and obligations that cannot be expressed in traditional policies. Examples of spatio-temporal policies include *"A volunteer is not allowed to access a patient's information for more than 10 minutes"* or *"A volunteer is not allowed to access a patient's symptoms if there is a doctor nearby."* Note that the latter policy must be dynamically instantiated to refer to specific elements in an event. That is, the rule should only apply to a volunteer and a doctor that are part of the same emergency call, thus belonging to the same event context. Rules that are not related to a specific event are described as *global*. The attribute $scope$ will distinguish between event and global policies.

An access authorization policy is expressed as a tuple:

$$P = \langle scope, subject, object, feature, granularity, ST \rangle$$

where

- $scope$ is the context in which the rule applies ($global$ or $event$).
- $subject$ denotes the entity (e.g. $volunteer$) to which the authorization is applied
- $object$ corresponds to the protected entity (e.g., patient whose medical record is accessed)
- $feature$ denotes the protected information (e.g., medical history, location, symptoms, diagnosis)
- $granularity$ captures the granularity of the spatio-temporal feature, or it can be $null$ (e.g., in the case of location information, we could specify an accuracy level of $neighborhood$ or $street$)
- $ST = \{ST_1, ..., ST_k\}$ represents the list of spatio-temporal constraints that control the policy

A spatio-temporal constraint represents a feature that has a spatial and/or a temporal dimension. Each spatio-temporal constraint ST_i can be mapped to the following schema:

$$ST_i = \{type, attribute, S, object, sign, time\}$$

The elements of the schema are defined as follows:

- Element $type$ is the type of the constraint. We consider range constraint RC, k-nearest neighbor constraint kNN, reverse k-nearest neighbor constraint $RkNN$, or $null$ if there is no spatial dimension.
- Element $attribute$ denotes an additional parameter of the constraint. For a range constraint, this represents travel time, and k for (reverse) k-nearest neighbor constraints.
- Element S represents the list of $subjects$ that participate in the constraint.
- Element $object$ corresponds to the subject, object, or location under consideration relative to S.
- Element $sign$ indicates if the constraint is $positive$ or $negative$.
- Element $time$ specifies the time validity (expiration), which varies within the interval $(0, \infty)$.

As an example of a policy, consider the rule, *"A volunteer is not allowed to access a patient's symptoms if there is a doctor in less than 10 meters."*. This policy would be expressed as the tuple

$$P = \langle \text{“}event\text{”}, \text{“}volunteer\text{”}, \text{“}patient\text{”}, \text{“}symptoms\text{”}, \phi, S_1 \rangle$$

The spatio-temporal constraint would be expressed as

$$S_1 = \langle \text{“}RC\text{”}, \text{“}10m\text{”}, \text{“}volunteer\text{”}, \text{“}doctor\text{”}, \text{“}negative\text{”}, \infty \rangle$$

Although the subjects in both the policy and the constraint are the same in this example, this will not always be the case. That is, to prevent redundant constraint expressions, S_1 could also be used in a policy tuple describing the doctor's permissions given that constraint.

3. SYSTEM ARCHITECTURE

3.1 Overview

In traditional access control systems, a user initiates a request and submits his or her relevant credentials, such as a username and password. The system then evaluates the combination of the subject, the object requested, the credentials provided, and the relevant policies, and it grants or denies access accordingly. In our proposed approach, a request is automatically generated in response to an event. The system then grants permissions to subjects, given the current spatio-temporal environment and policy constraints.

Our approach encompasses a number of key novelties. First, the system can grant permissions to multiple subjects simultaneously as part of a single request. Second, the subjects receiving access are not known when the request is initiated. That is, the system must determine the most appropriate subjects in response to a request. Finally, the dynamic nature of spatio-temporal environments necessitates continuous monitoring of subjects and objects. For example, if the traffic congestion increases in one location, the system may need to revoke or reassign authorizations in response to this. Despite these differences, we find that the traditional notions of Policy Decision Point (PDP) and Policy Enforcement Point (PEP) are still applicable.

In addition to the PDP and PEP components, we also introduce a Policy Database (PolicyDB) and a Moving Object Database (MOD). PolicyDB contains the access control policies as specified by Section 2.2. On the other hand, the positions of objects and subjects are indexed by the Moving Object Database, where the weights of the edges represent the driving time of each road segment (which forms a travel time network [4]). In our setting, where a timely response to an emergency is essential, the use of travel time is a better choice than the use of network distance, as the former metric captures and estimates the *time* that is required for an to travel to a certain location in the network. Moreover, we assume that the database is updated with real-time traffic information.

Figure 2 illustrates the interaction among the PEP/PDP and the rest of the system components when a new event is created (steps 1 and 2). As stated above, the PDP decides which entities will be granted access to the event information (step 3). In order to do this, the PDP queries the MOD to find subjects who have the required skills and are nearby the event location (the number and skills of the subjects that are required in each case are determined by the type of the event). Note that the MOD may need to provide more subjects than are required by the event type, in case one or more of the subjects is unable or unwilling to respond to the event. Then, the access policies involving those subjects are obtained by querying PolicyDB (step 4). As described in Section 2, these policies can contain static (organizational) and spatio-temporal rules that need to be continuously monitored by the PEP. Therefore, the PEP receives the subjects that will be involved in the event and the constraints (step 5). Finally, the monitoring of the constraints is initiated (step 6), and the subjects are notified (steps 7 and 8).

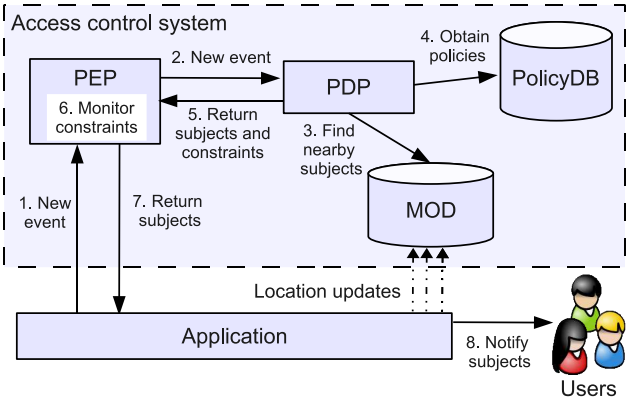


Figure 2: System Architecture

The Application layer abstracts the communication with the users, the sending of notifications, and the handling of location updates. In a variant of the scenario, the position of the users may be given by a external location provider.

3.2 Access Control Decision

The PDP needs to perform two main decision tasks. When an access request arrives in the system, it decides whether or not to grant the access based on the policies that refer to the subject. The second task, as mentioned previously, finds nearby and suitable subjects that may respond to a new event. The first operation corresponds to the functionality of a traditional access control system and is not considered in this paper.

For the latter, the PDP performs an incremental expansion of the network from the location of the event, using, e.g., the Incremental Network Expansion algorithm (INE) [7]. This algorithm retrieves the nearest neighbors from a starting point incrementally and ordered by distance to the point. We assume that the function $retrieve\_Next\_NN\_INE$ gets the next nearest neighbor that the INE algorithm would return.

In addition to the set of subjects, the PDP builds another set that corresponds to a back-up plan. This set consists of the subjects that will be deployed if any of the subjects in the primary plan fail to reach the emergency location. For example, a subject may not reply to the notification withing a reasonable time limit, or it may get delayed due to traffic. If this is the case, the system can quickly use the back-up plan to choose another subject without re-evaluating the entire request.

The pseudocode for the complete algorithm is shown in Figure 3.

The algorithm performs as follows. First, given the category of the event, the type and number of required entities is retrieved for both the primary and the backup plan (steps 2 and 3). These values are stored in pairs $[t_i, n_i]$, where t_i is a type and and n_i is the number of subjects of that type that are still required (i.e., this value will be at its maximum at the beginning of the execution, and it will decrease when subjects are found). The value k represents the maximum number of subject types in the system.

Then, INE is initialized (step 4). The algorithm executes while not all the subjects have been found (represented by the condition in step 5). For the next nearest neighbor that is found (step 6), the type is extracted (step 7). If there is a subject of that type required for either the primary plan (lines 8–10) or the back-up plan (lines 11–13), the subject is added to the corresponding set. The algorithm terminates when all the required subjects are found.

FindSubjects
Input: event location l and category c
Output: set of subjects for the primary plan S_p,
set of subjects for the back-up plan S_b
1. $S_p, S_b = \emptyset$
2. $p = \{[t_1, n_1]...[t_k, n_k]\} = entitiesPrimaryPlan(c)$
   /* t_i = subject type, n_i = number of subjects */
3. $b = \{[t'_1, n'_1]...[t'_k, n'_k]\} = entitiesBackupPlan(c)$
4. $initializeINE(l)$
5. **while** $\exists n_i > 0 \in p \vee \exists n'_i > 0 \in b$
   /* while more subjects are needed */
6. $s = retrieve\_Next\_NN\_INE()$ /* uses INE algorithm */
7. $t = typeOf(s)$
8. **if** $p[t].n > 0$ **then**
   /* if the plan needs a subject of that type, add it */
9. $S_p = S_p \cup \{s\}$
10. $p[t].n \leftarrow p[t].n - 1$
11. **else if** $b[t].n > 0$ **then**
   /* if the backup needs a subject of that type, add it */
12. $S_b = S_b \cup \{s\}$
13. $b[t].n \leftarrow b[t].n - 1$

Figure 3: Find Nearby Subjects

Recall that the distance used in $retrieve\_Next\_NN\_INE$ is measured as the expected time to travel from the current position to the location of the accident. Thus, the distance considered is the travel time distance instead of road network distance.

If the event has a $maxResponseTime$, the algorithm is modified to maintain a queue containing the discarded subjects (after line 13). In addition, the loop terminates when the distance (travel time) from the last nearest neighbor found is bigger than $maxResponseTime$. If no results are found when the expansion reaches $maxResponseTime$ or if the results are not enough, the entities belonging to the discarded queue are considered. We can assume that in these cases, the confidentiality can be relaxed in order to provide a fast response time. For example, a volunteer may be sent to the emergency instead of a doctor. Moreover, the system could create some range constraints that trigger an alert when some of the required subjects are in the vicinity.

We have illustrated a simplified version of the algorithm in which the roles are not interchangeable. That is, the algorithm would look for a volunteer even if there is a doctor nearby that can perform the same (and more) operations. If roles can be ordered or organized in a hierarchical manner, the algorithm should be modified to identify the subjects in a more efficient manner.

3.3 Constraint and Policy Processing

After the subjects that belong to the primary and the back-up plan are found, the PDP queries the PolicyDB to retrieve the policies affecting the subjects in both plans. As mentioned in Section 2.2,

some of these policies will need to be instantiated for the specific objects of the event. From the set of retrieved policies, the spatio-temporal constraints are extracted and sent to the PEP, which will start the monitoring process.

We identify two kinds of location constraints:

- nearest neighbor constraints (e.g., *"Volunteer A is a 2-nearest neighbor of the accident site"*)
- proximity (range) constraints (e.g., *"Volunteer A is within 5-minutes of the accident site"*)

Note that as the final decision regarding an access request is done by the PDP, both the PEP and the PDP need to maintain the list of active constraints with their state (however, recall that the active monitoring of the constraints is performed only by the PEP). For example, the policy, *"A volunteer is not allowed to access a patient's symptoms if there is a doctor nearby,"* will initially allow the volunteer to access the information, but will revoke the access when a doctor arrives. Similarly, the monitoring of the constraint of the policy, *"A volunteer is allowed to access a patient's symptoms when the arrival time to the patient is less than 1 minute,"* will allow the system to send a notification to the subject when s/he is close enough, without the subject requesting it. In other cases, for example if the system detects that a subject is delayed due to a traffic jam, the PDP may choose to send another subject to the emergency.

The PEP monitors the spatio-temporal constraints and notifies the PDP when a constraint changes state (from being met to not being met or vice versa). The factors that may change the state of a constraint, and thus need to be monitored, are the following:

1. movement of the subjects and/or the constraints
2. changes in the environmental conditions (e.g., traffic jams, represented as changes to the edge weights)
3. time expiration of the constraint
4. requirements of the event (e.g., change of the *category* of the event)
5. a subject becomes unavailable (subject changes status from *available* to *not available*)

The location constraints at the PEP can be monitored as described in the literature [6]. In this work, the authors propose two methods to monitor kNN's in road networks, the first of which maintains the query results by processing only updates that may invalidate the current NN sets, and the second of which follows the shared execution paradigm to reduce processing time. Moreover, the methods support object and query movement patterns and modifications of edge weights (thus supporting factors 1 and 2). Although the structures presented in this work can be applied in range queries and RkNN queries, solutions to efficiently support this functionality still need to be developed.

To monitor the time expiration (factor 3), the PEP can maintain a queue ordered by expiration time. When a constraint reaches its expiration time, it is removed from the system, and the PDP is notified.

For the last two factors (factors 4 and 5), the PEP can maintain two indices, based on the events (factor 4) and the subjects (factor 5). When these entities change status, all the constraints in which they are involved are eliminated from the system (and the PDP is notified as in the previous cases).

The PDP maintains a list of the active policies and constraints and the current state. This list is queried when a subject requests access to an object of the system. When a constraint changes its state, the factor that originated this change is examined. If the reason was a change of the event requirements (factor 4), the entire plan for the event needs to be re-evaluated. For the remaining cases, the system may make use of the back-up plan.

4. FUTURE WORK

In this position paper, we have identified several challenging research issues in location-based access control for helthcare emergency response. In the future, we plan to formalize the proposed event and policy models and to develop and prototype the functionality that must be implemented in the PEP and PDP. We also envision extending our proposal to take into account uncertainty in reporting location data, as well as the privacy issues that arise when subjects are required to report their location to the PEP.

Acknowledgments

The work reported in this paper has been partially supported by NSF grant 0712846 "IPS: Security Services for Healthcare Applications", and MURI award FA9550-08-1-0265 from the Air Force Office of Scientific Research.

5. REFERENCES

[1] S. Aich, S. Mondal, S. Sural, and A. K. Majumdar. Role based access control with spatiotemporal context for mobile applications. *Transactions on Computational Science IV: Special Issue on Security in Computing*, pages 177–199, 2009.

[2] S. Aich, S. Sural, and A. K. Majumdar. Starbac: Spatio temporal role based access control. In *OTM Conferences (2)*, volume 4804 of *Lecture Notes in Computer Science*, pages 1567–1582. Springer, 2007.

[3] C. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati. Access control in location-based services. In C. Bettini, S. Jajodia, P. Samarati, and S. Wang, editors, *Privacy in Location Based Applications*. Springer, 2009.

[4] W.-S. Ku, R. Zimmermann, H. Wang, and C.-N. Wan. Adaptive nearest neighbor queries in travel time networks. In *GIS '05: Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 210–219, New York, NY, USA, 2005. ACM.

[5] M. Kumar and R. E. Newman. Strbac - an approach towards spatio-temporal role-based access control. In *Communication, Network, and Information Security*, pages 150–155, 2006.

[6] K. Mouratidis, M. L. Yiu, D. Papadias, and N. Mamoulis. Continuous nearest neighbor monitoring in road networks. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 43–54. VLDB Endowment, 2006.

[7] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. Query processing in spatial network databases. In *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*, pages 802–813. VLDB Endowment, 2003.

Movement Data Anonymity through Generalization

[Position Paper]

Gennady Andrienko
Fraunhofer IAIS Sankt Augustin, Germany

Natalia Andrienko
Fraunhofer IAIS Sankt Augustin, Germany

Fosca Giannotti
KddLab ISTI-CNR Pisa, Italy

Anna Monreale
Computer Science Dept. University of Pisa

Dino Pedreschi
Computer Science Dept. University of Pisa

ABSTRACT

In recent years, spatio-temporal and moving objects databases have gained considerable interest, due to the diffusion of mobile devices (e.g., mobile phones, RFID devices and GPS devices) and of new applications, where the discovery of consumable, concise, and applicable knowledge is the key step. Clearly, in these applications privacy is a concern, since models extracted from this kind of data can reveal the behavior of group of individuals, thus compromising their privacy. Movement data present a new challenge for the privacy-preserving data mining community because of their spatial and temporal characteristics.

In this position paper we briefly present an approach for the generalization of movement data that can be adopted for obtaining k-anonymity in spatio-temporal datasets; specifically, it can be used to realize a framework for publishing of spatio-temporal data while preserving privacy. We ran a preliminary set of experiments on a real-world trajectory dataset, demonstrating that this method of generalization of trajectories preserves the clustering analysis results.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Spatial databases and GIS; K.4.1 [**Public Policy Issues**]: Privacy

Keywords

k-anonymity, Privacy, Spatio-temporal, Clustering

1. INTRODUCTION

Many Knowledge Discovery techniques have been developed that provide new means for improving personalized services through the discovery of patterns which represent typical or unexpected customer's and user's behavior. However, the collection and the disclosure of personal, often sensitive, information increase the risk of citizen's privacy violation. For this reason, many recent research works have focused on privacy-preserving data mining [2, 13, 6, 7]. In general, these approaches allow to extract knowledge while trying to protect the privacy of individuals represented in the dataset. Some of these techniques return anonymized data mining results, while others provide anonymized datasets to the companies/research institution in charge of their analysis. The pervasiveness of location-aware devices, e.g., PDAs and cell phones with GPS technology, RFID devices enables to collect a great amount of traces left by moving objects and to analyze their motion patterns. Clearly, in this context privacy is a concern: location data allows inferences which may help an attacker to discovery personal and sensitive information like habits and preferences of individuals. Hiding car identifiers for example replacing them with pseudonyms as shown in [13] is insufficient to guarantee anonymity, since location represents a property that could allow the identification of the individual. In particular, sensitive information about individuals can be uncovered with the use of visual analytics methods. Therefore, in all cases when privacy concerns are relevant, such methods must not be applied to original movement data. The data must be anonymized, that is, transformed in such a way that sensitive private information could no more be retrieved.

In this position paper we present a method for the generalization of movement data that can be adopted for obtaining a form of anonymity in spatio-temporal datasets. The main idea is to hide locations by means of generalization, specifically, replacing exact positions in the trajectories by approximate positions, i.e. points by areas. This method of generalization can be used in a privacy-preserving framework of spatio-temporal data in order to generate an anonymous dataset which satisfy the k-anonymity property. In the literature, most of anonymization approaches proposed in the spatio-temporal context are based on randomization techniques, space translations of points and suppression of some potions of a trajectory. To the best of our knowledge only the work in [15] uses spatial generalization to achieve anonymity for trajectory datasets; however, a fixed grid hierarchy is used in this work to discretize the spatial dimension. In contrast, the novelty of our approach lies in finding a suitable tessellation of the territory into areas depending of the input trajectory dataset. As a result of our approach, we obtain anonymous trajectories with high analytical utility, if compared with previous works (both randomization based and generalization based): in particular, we show how the results of clustering analysis are faithfully preserved. The con-

ACM SPRINGL '09 November 3, 2009. Seattle, WA, USA

cept of spatial generalization has been also used in the works on privacy in location-based services [5, 8, 9], where the goal is on-line anonymization of individual location-based queries, while our aim is privacy-preserving data publishing, which requires the anonymization of each entire trajectory. A detailed discussion appears in Section 2.

The rest of the paper is organized as follows. Section 2 discusses the relevant related works on privacy issue in spatio-temporal data. Section 3 introduces the problem definition. In Section 4 we describe the generalization approach for trajectories. Section 5 describes the experimental results on the clustering analysis. In Section 6 we discuss about the possible ideas to adapt the proposed generalization method to anonimyze movement data. Finally, Section 7 concludes.

2. RELATED WORK

Many research works have focused on techniques for privacy-preserving data mining [2] and for privacy-preserving data publishing. The first operation before data publishing is to replace personal identifier with pseudonyms. In [13] authors showed that this simple operation is insufficient to protect privacy. In this work, Samarati and Sweeney propose k-anonymity to make each record indistinguishable with at least $k-1$ other records. k-anonymity is the most popular method for the anonymization of spatio-temporal data. It is often used both in the works on privacy issues in location-based services (LBSs) and on anonymity of trajectories.

In LBSs context a trusted server usually has to handle the requests of users and to pass them on to the service providers. In general, it has to provide a on-line service without compromise the anonymity of the user. The different systems proposed in literature to make the requests indistinguishable from $k-1$ other requests use a space generalization, called *spatial-cloaking* [5, 8, 9]. In our context the anonymization process is off-line, as we want to anonymize a static database of trajectories. To the best of our knowledge only three works address the problem of k-anonymity of moving objects by a data publishing perspective [1, 10, 15]. In the work [1], the authors study the problem of privacy-preserving publishing of moving object database. They propose the notion of (k, δ)-anonymity for moving objects databases, where δ represents the possible location imprecision. In particular, this is a novel concept of k-anonymity based on co-localization that exploits the inherent uncertainty of the moving objects whereabouts. In this work authors also propose an approach, called *Never Walk Alone* based on trajectory clustering and spatial translation. In [10] Nergiz et al. address privacy issues regarding the identification of individuals in static trajectory datasets. They provide privacy protection by: (1) first enforcing k-anonymity, meaning every released information refers to at least k users/trajectories, (2) then reconstructing randomly a representation of the original dataset from the anonymization. Yarovoy et al. in [15] study problem of k-anonymization of moving object databases for the purpose of their publication. They observe the fact that different objects in this context may have different quasi-identifiers ans so, anonymization groups associated with different objects may not be disjoint. Therefore, a novel notion of k-anonymity based on spatial generalization is provided. In this work, authors propose two approaches in order to generate anonymity groups that satisfy the novel notion of k-anonymity. These approaches are called *Extreme Union* and *Symmetric Anonymization*.

Another approach based on the concept of k-anonymity is proposed in [11], where a framework for k-anonymization of sequences of regions/locations is presented. The authors also propose an approach that is an instance of the proposed framework and that allows to publish protected datasets while preserving the data utility for sequential pattern mining tasks. This approach, called *BF-P2kA*, uses a prefix tree to represent the dataset in a compact way. Given a threshold k generates a k-anonymous dataset while preserving the sequential pattern mining results.

Finally, in a very recent work [14], a suppression-based algorithm is suggested. Given the head of the trajectories, it reduces the probability of disclosing the tail of the trajectories. This work is based on the assumption that different attackers know different and disjoint portions of the trajectories and the data publisher knows the attacker knowledge. So, the solution is to suppress all the dangerous observations.

3. PROBLEM DEFINITION

A moving object dataset is a collection of trajectories $D = \{T_1, T_2, \ldots, T_m\}$ where each T_i is a trajectory represented by a sequence of spatio-temporal points:

$$T_i = (x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)$$

and $(t_1 < t_2 < \cdots < t_n)$.

Given a moving object dataset D our goal is to provide an anomymized version of D that guarantees the privacy of the individuals while preserving some interesting analysis results such as clustering analysis. For this aim we want to use a k-anonymity approach based on spatial generalization.

In this position paper we describe a method for generalization of movement data that can be adapted for obtaining anonymity in a moving object dataset. The idea is to hide personal information by means of generalization, specifically, replacing exact positions in the trajectories by approximate positions, i.e. points by areas.

4. TRAJECTORY GENERALIZATION

The approach to generalize movement data consists of two main steps: (1) generating a division of the territory into areas and (2) generalizing the original trajectories.

4.1 Division of the Territory into Areas

The generalization method generates an appropriate division of the territory into areas. The method is based on extracting characteristic points from the trajectories, which are the positions of start and end, significant turns (i.e. the change of the movement direction is above a given threshold), and significant stops (i.e. the time of staying in the same position is above a threshold). The extracted points are grouped into spatial clusters. The central points of the clusters are used as generating points for Voronoi tessellation of the territory, which produces suitable areas. Since the areas are built around clusters of characteristic points, the resulting abstraction conveys quite well the principal characteristics of the movement. The level of the abstraction can be controlled through the parameters of the clustering method.

We demonstrate the work of the method by example of a subset car trajectories got by the European project GeoPKDD. Thanks to this project we received a real-world and

large dataset of trajectories of cars equipped with GPS and moving in the city of Milan (Italy). For our preliminary experiments we considered 4287 trajectories. Figure 1(a) presents a map with the original trajectories. In Figure 1(b), there are the characteristic points extracted from the trajectories (54362 points in total). Both the trajectories and the characteristic points are shown with 10% opacity, to enable the estimation of the densities in different places. In Figure 1(c), the characteristic points have been clustered. The clusters are represented by colouring. We use a special spatial clustering algorithm with a parameter defining the maximum radius (spatial extent) of a cluster. The clusters in Figure 1(c) have been obtained for the value 500 metres of this parameter. Figure 1(d) presents the centroids of the point clusters and the Voronoi cells, which have been built using the centroids as generating points. Besides the cluster centroids, we add generating points around the boundaries of the territory and in the areas where there are no characteristic points from the trajectories. This is done for the cells to be more even in sizes and shapes.

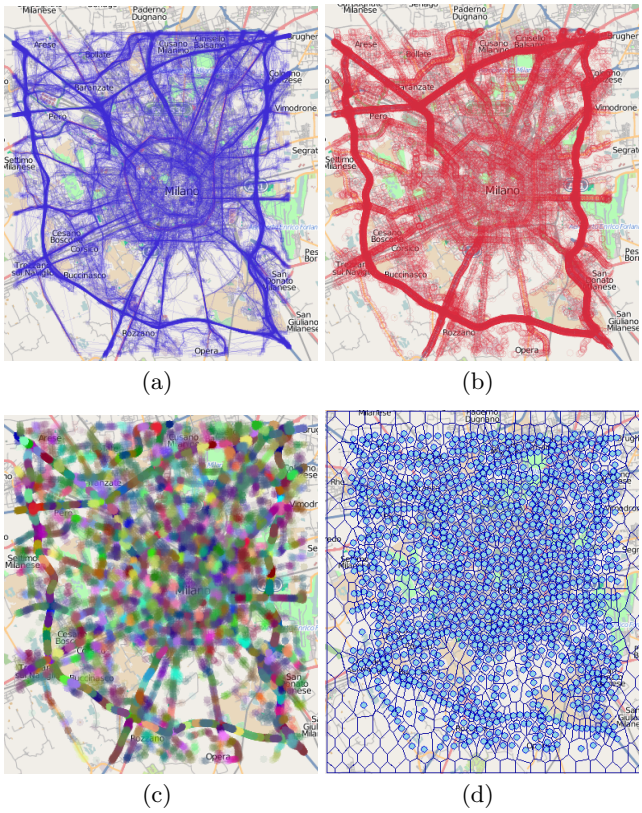


Figure 1: a) Original subset of 4287 trajectories (10% opacity). b) The characteristic points extracted from the trajectories (10% opacity). c) Clustered characteristic points. d) Centroids of the clusters and the Voronoi tessellation of the territory.

4.2 Generalization method

After obtaining the division of the territory, the trajectories are generalized in the following way. We apply place-based division of trajectories into segments. For each trajectory, the area a_1 containing its first point p_1 is found. Then, the second and following points of the trajectory are checked for being inside a_1 until finding a point p_i not contained in a_1. For this point p_i, the containing area a_2 is found.

The trajectory segment from the first point to the i-th point is represented by the vector (a_1, a_2). Then, the procedure is repeated: the points starting from p_{i+1} are checked for containment in a_2 until finding a point p_k outside a_2, the area a_3 containing p_k is found, and so forth up to the last point of the trajectory.

In the result, the trajectory is represented by the sequence of areas $\{a_1, a_2, \ldots, a_n\}$. There may be also a case when all points of a trajectory are contained in one and the same area a1. Then, the whole trajectory is represented by the sequence $\{a_1\}$. For each area a_i in the sequence, there is a corresponding time interval starting with the time moment of the first position in a_i and ending with the time moment of the last position in a_i.

As most of the methods for analysis of trajectories are suited to work with positions specified as points, the sequence of areas $\{a_1, a_2, \ldots, a_n\}$ is replaced, for practical purposes, by the sequence $c_1, c_2, \ldots, c_n$ consisting of the centroids of the areas $\{a_1, a_2, \ldots, a_n\}$. As a result, we obtain generalized trajectories. Figure 2 illustrates the generalized trajectories of the cars from Milan.

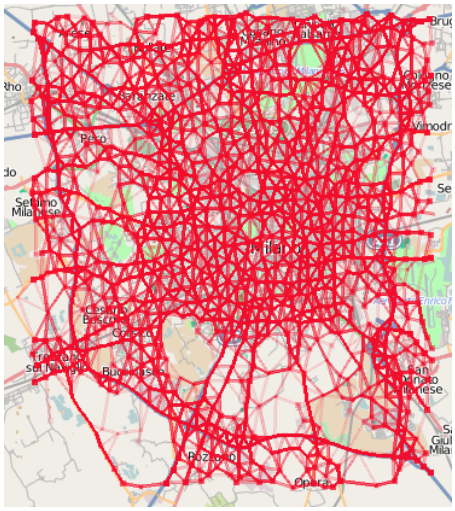


Figure 2: Generalized trajectories of cars from Milan (10% opacity); the line thickness is 2 pixels.

5. CLUSTERING ANALYSIS

An important property of this method for protecting personal data is that the resulting transformed data are suitable at least for some kinds of analysis. In particular, it is possible to analyze the flows between the areas and statistics of the visits of the areas. One may also analyze the statistics of the travel times between different pairs of areas, not only neighboring. Frequently occurring sequences of visited areas can be discovered by means of data mining techniques. It is also possible to apply cluster analysis to the modified trajectories. Thus, we have made several experiments with clustering of the original car trajectories from Milan and generalized versions of these trajectories using the generic density-based clustering algorithm *OPTICS* [4] with a suitable distance function.

We found that the results of clustering the original and the generalized trajectories are very similar when the distance threshold (the parameter of the clustering algorithm) for the generalized trajectories is about one half of the distance threshold for the original trajectories. Figure 3 shows the biggest clusters obtained from the original set of trajectories (the first group of 12 clusters in the figure) and the biggest clusters obtained from the set of generalized trajectories (the last group of 12 clusters). The clusters are represented in an aggregated form. The results have been obtained using the density-based clustering algorithm OPTICS with the distance function "*route similarity*" [3, 12] and the required minimum of 5 neighbors of a core object. The distance thresholds used is 500m for the first group and 250m for the second group. The labels A, B, etc. establish the correspondence between the clusters in two results. The clusters of the second group corresponding to the clusters G and L of the first group are not among the largest 12 clusters (they are on the 15th and 14th places, respectively). Analogously, the clusters of the first group corresponding to the 11th and 12th clusters of the second group (clusters with label N and Q in the figure) are on the 14th and 17th places, respectively.

The results of the experiments allow us to believe that the idea has a good potential.

6. GENERALIZATION VS K-ANONYMITY

The approach described in Section 4, given a dataset of trajectories allows us to generate a generalized version of it. In order to adapt this method to the anonymization of movement data, some extensions are required. In particular, it is necessary to ensure that:

1. each area contains positions from the trajectories of k different people, where k is a parameter.
2. the dispersion of the positions in each area is not less than a specified threshold (another parameter).
3. for each pair of areas a and b there are either none or at least k people who come from a to b (possibly, with visiting some other areas in between).

The satisfaction of these anonymity conditions is easy to check. Thus, the map in Figure 4 visualizes the numbers of different trajectories that visited the areas of the territory division (Voronoi cells) used for the generalization. The cells where the first two conditions are not satisfied must be enlarged to include more positions. This is done by producing a new Voronoi tessellation after excluding the generating points of the "problematic" cells. Similarly, when too few people come from a to b, either a or b is excluded. To choose between a and b, the total number of incoming and outgoing links with the magnitudes below k is counted for each of them. Excluded is the area where this number is bigger.

The generalization-based anonymization method is currently under development. We need to do further investigations for checking whether any risks to personal privacy are indeed precluded when trajectories are anonymized in this way.

7. CONCLUSION

In this position paper, we present an approach for generalization of movement data. We think that this method

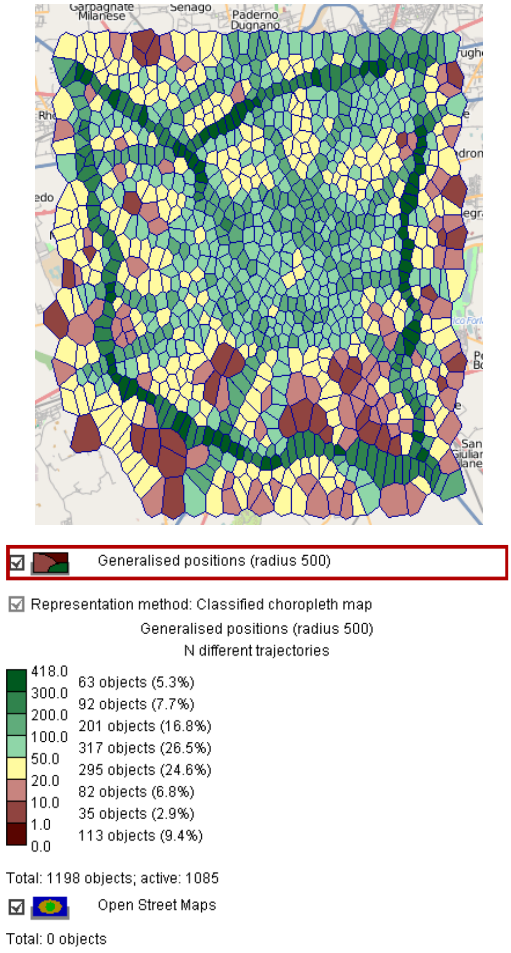


Figure 4: The map shows the numbers of different trajectories that visited the areas of the territory division by area coloring. The areas that do not contain any points from the trajectories are hidden.

can be adopted to realize a framework for anonymization of spatio-temporal data based on spatial generalization and the k-anonymity concept. Through a preliminary set of experiments on a real-life mobility dataset, we showed that the proposed technique preserves clustering results.

In future work, we intend to investigate further the protection model against the re-identification attack, that can be obtained using the method proposed in this paper.

8. REFERENCES

[1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *ICDE*, pages 376–385, 2008.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD*, pages 439–450. ACM, 2000.

[3] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. In *VAST in press*, 2009.

[4] M. Ankerst, M. M. Breunig, H.P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *SIGMOD*, pages 49–60, 1999.

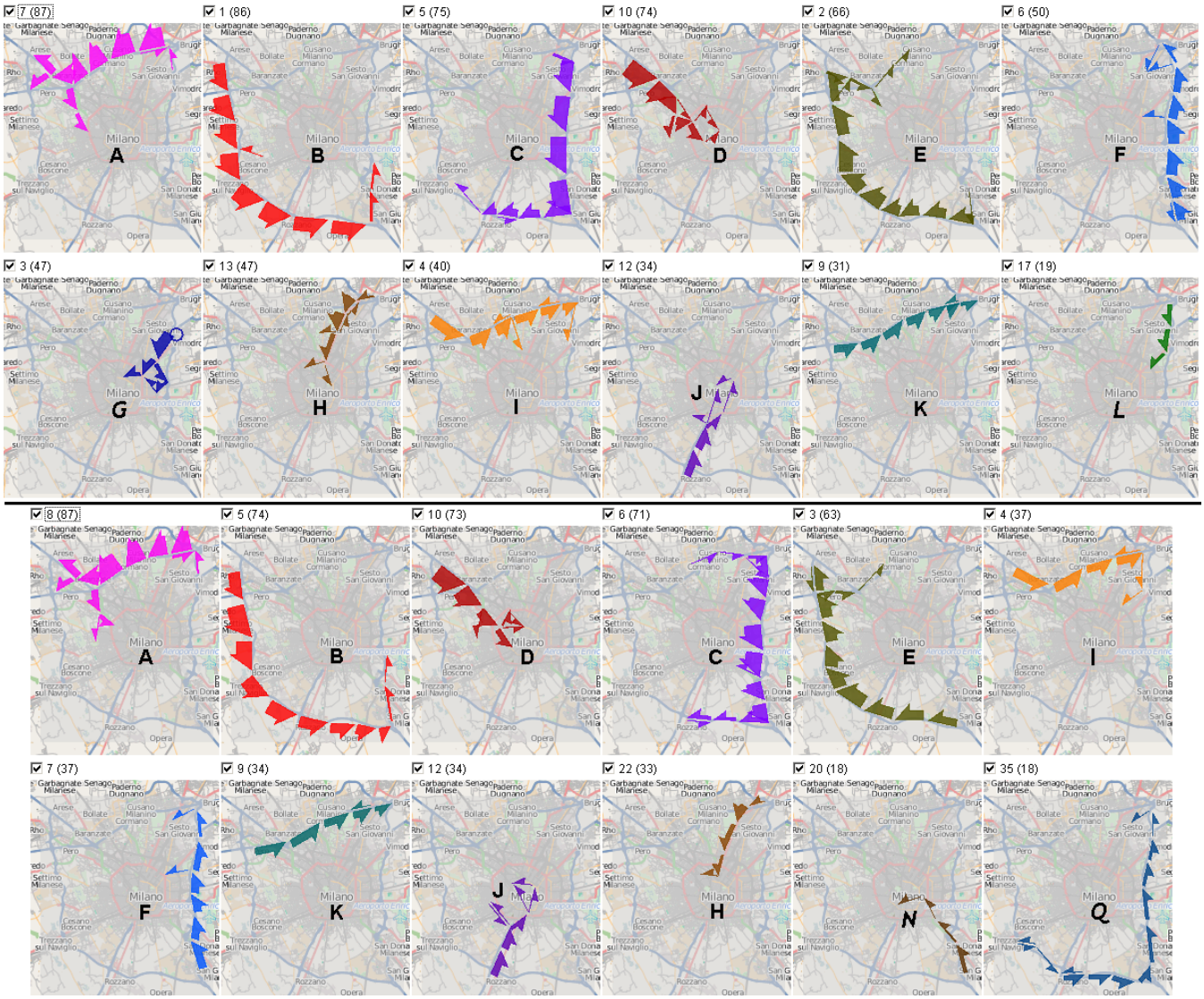


Figure 3: Comparison of clustering results for the original (top) and generalized (bottom) trajectories. 12 biggest clusters from each result are visible.

[5] M. Gruteser and D. Grunwald. A methodological assessment of location privacy risks in wireless hotspot networks. In *SPC*, pages 10–24, 2003.

[6] R. J. Bayardo Jr. and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE*, 2005.

[7] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *ICDE*, page 25, 2006.

[8] M. F. Mokbel, C. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *VLDB*, pages 763–774, 2006.

[9] M. F. Mokbel, C. Chow, and W. G. Aref. The new casper: A privacy-aware location-based database server. In *ICDE*, pages 1499–1500, 2007.

[10] M. E. Nergiz, M. Atzori, and Y. Saygin. Perturbation-driven anonymization of trajectories. Technical Report 2007-TR-017, ISTI-CNR, Pisa, 2007.

[11] R. G. Pensa, A. Monreale, F. Pinelli, and D. Pedreschi. Pattern-preserving k-anonymization of sequences and its application to mobility data mining. In *PiLBA*, 2008.

[12] S. Rinzivillo, D. Pedreschi, M. Nanni, F. Giannotti, N. Andrienko, and G. Andrienko. Visually-driven analysis of movement data by progressive clustering. *Information Visualization*, 7(3/4):225–239, 2007.

[13] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International, 1998.

[14] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *MDM*, pages 65–72, 2008.

[15] R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, and W. H. Wang. Anonymizing moving objects: how to hide a mob in a crowd? In *EDBT*, pages 72–83, 2009.

Protecting location privacy against spatial inferences: the PROBE approach

Maria Luisa Damiani
Universitá degli Studi, Milano, I
damiani@dico.unimi.it

Elisa Bertino
Purdue University, USA
bertino@cs.purdue.edu

Claudio Silvestri
Universitá di Venezia, I
silvestri@dsi.unive.it

ABSTRACT

The widespread adoption of location-based services (LBS) raises increasing concerns for the protection of personal location information. A common strategy, referred to as obfuscation, to protect location privacy is based on forwarding the LSB provider a coarse user location instead of the actual user location. Conventional approaches, based on such technique, are however based only on geometric methods and therefore are unable to assure privacy when the adversary is aware of the geographical context. This paper provides a comprehensive solution to this problem. Our solution presents a novel approach that obfuscates the user location by taking into account the geographical context and user's privacy preferences. We define several theoretical notions underlying our approach. We then propose a strategy for generating obfuscated spaces and an efficient algorithm which implements such a strategy. The paper includes several experimental results assessing performance, storage requirements and accuracy for the approach. The paper also discusses the system architecture and shows that the approach can be deployed also for clients running on small devices.

Categories and Subject Descriptors

H.2.0 [**General**]: Security, integrity, and protection; H.2.8 [**Database Management**]: Spatial Databases and GIS

General Terms

Design, Experimentation, Security

Keywords

Location Privacy, Location Based Services

1. INTRODUCTION

The ever increasing collection of personal location data, pushed by the widespread use of location-sensing technologies, like satellite positioning systems, RFID and sensors, and the development of location-based services (LBS), motivates the great concern for the protection of personal location information (*location privacy*). The communication of a user's position to a LBS provider upon a service request may result in the unauthorized dissemination of personal location data. Such data, combined with other available information, may in turn lead to the inference of sensitive information about individuals. Various approaches have been thus proposed to assure location privacy. Most of those approaches are based on obfuscation techniques that aim at disguising the actual position of the user by forwarding to the LBS provider fake or less accurate (*generalized*) location information. Approaches based on k-anonymization refine obfuscation techniques by making sure that the generalized location of each user is indistinguishable from the generalized locations of other $k-1$ users.

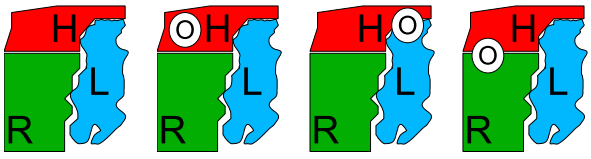


Figure 1: Geographical context

A common problem to all the above approaches is that they do not take into account geographical knowledge that the adversary may have about the reference spatial context. We claim that by exploiting such knowledge, the adversary may be able to obtain more precise bounds about the actual user location (referred to as *location inference*), thus defeating the obfuscation mechanism. Another major drawback of such approaches is that they do not support location privacy preferences, that is, the specification of which locations are sensitive for which users. Not all locations may have the same sensitivity for all the users and therefore a suitable obfuscation mechanism should be able to generate obfuscation locations that are tailored to the privacy preference of each user. We believe that, as we move toward more personalized LBS, privacy should be one of key personalization dimensions. Before moving to introduce the key contribution of the paper, we introduce a running example to illustrate the location inferences that are possible when geographical knowledge is available to the adversary.

EXAMPLE 1. *Assume that a user of a LBS is located within a hospital which for this user is a sensitive place. Consider the geographical context in Figure 1.a: the hospi-*

ACM SPRINGL '09 November 3, 2009. Seattle, WA, USA

tal H is close to a lake L and to a residential district R; all these places, i.e., the lake, the district and the hospital, cover a polygonal region. Suppose that no boats are allowed on the lake and that the adversary has this knowledge. Assume also that the actual position of the user is obfuscated by region O containing the user's position (obfuscated location). From the observation of the spatial relationships existing between the obfuscated location and the spatial entities, like spatial containment, overlaps and disjointness, the adversary can easily infer whether the user is located in a sensitive place. In particular consider the following three cases:

i) The obfuscated location is spatially contained in the extent of the hospital (Figure 1.b). In this case, the adversary may easily infer that the user is located in a sensitive place, that is, the hospital, although the actual position is blurred to a coarser region.

ii) The region corresponding to the user's obfuscated location includes the extent of both the hospital and the lake (Figure 1.c). Since the user cannot be physically located inside the lake, because no boats are allowed on the lake, the only realistic position is within the hospital and thus the obfuscated location is still sensitive. Notice that in this case information about the user's obfuscated location is combined with publicly available information, i.e., that no boat is allowed on the lake, in order to infer more precise information about the actual location of the user.

iii) The region corresponding to the user's obfuscated location overlaps part of the hospital and part of the residential district (Figure 1.d). Since the hospital is the only sensitive place, we can say that the obfuscated location is "sensitive to some extent".

Suppose now that the user is a physician. In such case the fact of being located in the hospital is likely not to be sensitive for this user.

The example emphasizes the fact that a location, besides a geometric shape, has a semantics (e.g., hospital) which depends on the entities spatially related to such position. The example clearly shows that privacy breaches occur because existing obfuscation techniques are unable to protect against the inferences made by linking the geometric information with the *semantic location* which, depending on the perceptions of users, may represent sensitive information. The protection of sensitive location information thus requires techniques able to take into account the geographical context in which users are located, in particular the semantic locations and the spatial distribution of population, as well as the users' privacy preferences. To our knowledge, a comprehensive approach to this problem has not been investigated yet.

In this paper we take a step in that direction and present a novel method for the personalized obfuscation of semantic locations. A key concept in our approach is the **sensitivity metric** which quantifies the sensitivity of a region, i.e., "how much private" a region is. The choice of the metric is crucial, and, indeed, different metrics can be devised. We define the sensitivity of a region r with respect to a certain category of semantic locations (e.g., hospitals, religious buildings) as the probability that a user in r is inside any place of that category. Users can specify in a **profile** which categories of semantic locations are sensitive as well as the desired degree of protection for each of those categories. We define the user profile as a set of constraints on the maximum sensitivity of obfuscated locations which is tolerated by the user. The privacy-preserving strategy is then articulated in two stages: in the first stage, an **obfuscation algorithm** generates a set of regions (*obfuscated locations*) masking the extent of sensitive locations. Each of those regions includes both sensitive and innocuous semantic locations, and satisfies the user profile constraints. In the second stage, the user's position is checked against the obfuscated locations and, if the user falls inside location r, then r is disclosed in place of the exact position. It is important to observe that the obfuscation algorithm does not take into account the user's position. This way, an attacker cannot exploit the knowledge of the algorithm to infer more precise bounds over the user's position inside the larger region. Therefore the method is robust against reverse engineering attacks.

The above elements are combined in the framework, referred to as PROBE (Privacy-preserving Obfuscation Environment). The PROBE framework can be flexibly deployed on either a two-tier architecture or in alternative, whenever the client devices have limited capabilities, on a three-tier architecture [5]. In the former case, the obfuscated locations are generated by the client; in the latter case, by a third system, e.g., a Web application or a laptop, and then downloaded on the client. To summarize, the key contributions of this paper are:

- A privacy model for the specification of privacy preferences on semantic locations. Semantic locations are defined in compliance with geo-spatial standards. The privacy model comprises the sensitivity metric and the user profile model.
- An obfuscation algorithm called $Sens_{Hil}$. Experimental results show that our algorithm is very efficient and the size of obfuscated maps is very small and thus suitable for storage on small devices.

The rest of the paper is organized as follows. Next section overviews related work. Then we introduce the location privacy model. The obfuscation algorithm is described in the subsequent section followed by the experimental evaluation. Final remarks and future plans are finally reported in the concluding section.

2. RELATED WORK

The bulk of research on location privacy in LBS has focused on the development of spatial cloaking techniques [3, 21, 4, 7, 12, 9, 10]. In place of the exact user's position, spatial cloaking methods disclose a less accurate cloaked region CR containing the user. Spatial cloaking is extensively applied to provide spatial k-anonymity [4, 7, 12, 9, 10]. The exact position of an individual is replaced by a CR containing k users, while the privacy metric is defined by the probability $1/k$ of identifying a user in CR. However, spatial k-anonymity does not provide any protection against location inferences [1]. For example, if an attacker knows that John is among k users in the cloaked region CR, and CR is inside a hospital, then the attacker can promptly infer that John might have health problems. The reason of this privacy leak is that spatial k-anonymity only protects users' identities and not the semantic locations.

A different approach which provides strong privacy guarantees is to encode a location-based query using cryptographic techniques. For example, Ghinita et al. [6] present a technique based on the PIR theory (Private Information Retrieval) to compute nearest-neighbor (NN) queries without disclosing any location information about the user and the requested points of interest. This technique, however, suffers from two main drawbacks: it is tailored on a particular class of queries, i.e., NN-queries; furthermore, it incurs in high communication costs.

The protection of semantic locations has been addressed by Xue et al. [20]. The goal is to prevent the individuals inside a spatial k-anonymous region from being located in a semantic location. The method extends the notion of *l-diversity* [11] to the spatial case by adding the privacy constraint that a k-anonymous region must contain not only k users, but also l different semantic locations. This approach, however, does not take into account the personal preferences of users (i.e., the hospital is sensitive for a patient, but it is not so for a doctor). Further the resulting region can be very broad, especially in geographically homogeneous areas. Our approach overcomes those drawbacks and provide a flexible solution in which users can request personalized obfuscation while limiting the loss of spatial resolution.

Personalized privacy preservation is the major goal of policy-based approaches. Location privacy policies, in particular, are commonly applied to control the disclosure of personal locations to third parties [14, 22, 8, 18, 16]. Those polices rely on models and protocols which often derive from the W3C Platform for Privacy preferences (P3P) like in [14] or that are rooted in access control [18, 22, 8, 16]. Those approaches, however, are not able to contrast inference channels, and in particular location inferences. On the other hand, privacy personalization in the framework of spatial k-anonymity [4] allows the specification of very simple parameters, like the value of k and the minimum size of the cloaked region. More pertinent to our problem, is the approach by Tao et al. [19] presenting a privacy personalization technique for the protection of sensitive attributes in non-spatial, k-anonymous datasets. The substantial difference with PROBE, is that our system not only allows the specification of user preferences to control location inferences but also is able to generate generalized locations.

To summarize, there are various proposals which aim to protect location either through spatial cloaking, policy-based approaches or using strong but expensive solutions like cryptographic techniques. None of them, however, is able to provide at the same time a personalized and cost-effective protection of semantic locations, which instead is the unique contribution of PROBE.

3. THE PRIVACY MODEL

We now introduce the privacy model defined in PROBE. The model is articulated in four components, namely the space model, the sensitivity metric, the user profile, and the obfuscated location model, that we describe in detail in what follows.

3.1 Space model

The region of concern for the LBS application is referred to as *reference space* Ω. Ω is a possibly bounded and connected area in a two-dimensional space. The *geometric objects* in Ω have a spatial type compliant with geo-spatial standards [15]. The user's position is a point. Moreover spatial types are closed under (appropriately defined) geometric union $\cup^s$, intersection $\cap^s$, difference ($\setminus^s$).

Following geo-spatial standards, the semantic locations are described in terms of *spatial features* (simply *features* hereinafter). Features have a type. Further features have an extent of region type. Moreover, we assume features to be spatially disjoint. Note that if two places are one contained in the other, the corresponding features must be defined so that they do not overlap. For example, if a restaurant is within a park, the extent of the park feature should have a hole corresponding to the extent and location of the restaurant feature. We denote with $Cov(ft)$ a function which yields the spatial union of the extents of all features of type ft. The pair of sets (FT, F) representing respectively feature types and features is referred to as the *geographical database* of the application.

Feature types can be classified as sensitive. The classification easily extends to regions. Let r be a region and F_S be the set of sensitive feature types. We say that r is *sensitive* if it overlaps $Cov(ft)$, for some sensitive feature type ft, that is,

$$\bigcup_{ft \in FT_S}^{s} Cov(ft) \cap^s r \neq \emptyset.$$

3.2 Sensitivity metric

The distribution of the user's positions in the reference space is assumed to be known and is described by the probability density function *pdf*. $P(r) = \int_r pdf$ denote the probability that a user, known to be located in Ω, is actually located in region r; $P(\Omega) = 1$ and $P(\emptyset) = 0$. We say that a region r is *unreachable* if $P(r) = 0$; instead r is *reachable* if at least one subregion $r' \subseteq r$ exists such that $P(r') > 0$. We refer to such a probability as *user's position probability*.

The user's position probability is used to define the sensitivity of a region. The sensitivity of a region is a value in the interval $[0, 1]$ which quantifies "how much private" a region is. Consider a sensitive feature type ft. We define *sensitivity of r wrt ft*, denoted as $P_{sens}(ft, r)$, the probability that a user, known to be in r, is actually within the extent of any sensitive feature of type ft overlapping with r. Such a sensitivity is expressed in terms of conditional probability as follows:

$$P_{sens}(ft, r) = \begin{cases} P(Cov(ft)|r) & \text{if} P(r) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Note that the sensitivity of an unreachable region is set to 0 for any feature type because a user cannot be located in such a region. Further, if the region is entirely covered by sensitive features of the same type ft_o, then $P_{sens}(ft_o, r) = 1$. The function $P_{sens}(ft, r)$ can be rewritten as:

$$P_{sens}(ft, r) = \begin{cases} \frac{\int_{Cov(ft) \cap^s r} pdf}{\int_r pdf} & \text{if } \int_r pdf \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

EXAMPLE 2. *Consider the region r in Figure 2. Such a region overlaps two features H_1 and H_2 of type Hospital, which is sensitive feature type. H_1 is partially contained in r and H_2 is entirely inside the region. Moreover the region includes lake L. Assume the following distribution of user's positions: L is unreachable; the user's positions in $r \setminus^s L$*

are equally probable. The sensitivity of r with respect to the feature type Hospital can be computed as follows:

$$P_{sens}(Hospital, r) = \frac{Area(H_1 \cap^s r) + Area(H_2)}{Area(r \setminus^s L)}$$

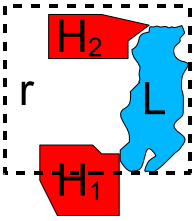


Figure 2: Example of sensitive region including an unreachable region

3.3 The privacy profile

The privacy profile specifies the set of sensitive feature types, i.e., FT_S and the set of user preferences. A privacy preference (*preference* for short) is a constraint over the maximum sensitivity that the user tolerates in any location wrt a certain sensitive feature type in FT_S. Given a feature type $ft \in FT_S$, a preference takes the form

$$\mathcal{T}(ft) = v$$

where $v \in (0, 1)$ is the *threshold sensitivity value* of ft. We say that a region r *satisfies* preference $\mathcal{T}(ft) = v$ if the following inequality holds:

$$P_{sens}(ft, r) \leq v.$$

Note that we do not consider the preference $\mathcal{T}(ft) = 1$ because it would mean that ft is not sensitive, against the initial assumption. We also rule out the preference $\mathcal{T}(ft) = 0$ because it can be only satisfied if ft has no instances which is not an interesting case.

The privacy profile takes the form of the tuple:

$$< FT_S, T >$$

where $FT_S = \{ft_1, ..ft_n\}$ is the set of sensitive features types and $T = \{\mathcal{T}(ft_1), ...\mathcal{T}(ft_n)\}$ is the set of preferences, one for each sensitive feature type.

EXAMPLE 3. *A possible privacy profile of a user who is concerned with the disclosure of positions in religious buildings and in health organizations can be defined as follows:*

- $FT_S = \{HealthOrganization, ReligiousBuilding\}$
- $T = \{\mathcal{T}(hospital) = 0.4,\ \mathcal{T}(ReligiousBuilding) = 0.1\}$.

It can be noticed that the threshold value is lower for the feature type ReligiousBuilding *than for the feature type HealthOrganization to mean that the privacy demand is stronger in the former case.*

3.4 Obfuscated location and obfuscated map

At this point, we are able to formally define the concept of *obfuscated location*. Consider the privacy profile $p =< F_S, T >$ with $FT_S = \{ft_1...ft_n\}$. Let r be a region. We say that r is an *obfuscated location* for profile p if and only if r is sensitive and every preference in the set T is satisfied. The latter property can be formally expressed as:

$$\forall ft \in FT_S, P_{sens}(ft, r) \leq \mathcal{T}(ft)$$

where $T = \{\mathcal{T}(ft)\}_{ft \in FT_S}$ are the privacy preferences of the profile p. A related concept is that of *obfuscated map*. An obfuscated map for profile p is a set $S = \{r_1, ..r_n\}$ of obfuscated locations which are *disjoint* and *cover* the whole set of sensitive features. Since the regions are disjoint, a user can be located at most in one obfuscated location. We recall that the obfuscated locations are disjoint if their spatial intersection is the empty set, that is: $\forall i, j \in \{1..n\}, i \neq j \Rightarrow r_i \cap^s r_j = \emptyset$. Moreover, since the regions cover the sensitive features, every sensitive position falls inside an obfuscated location. Such a condition is verified if the spatial union of the obfuscated locations is a superset of the sensitive portion of space, that is:

$$\bigcup_{i \in \{1..n\}}^{s} r_i \supset \bigcup_{ft \in FT_S}^{s} Cov(ft)$$

Note that the obfuscated map does not cover necessarily the whole space.

4. THE OBFUSCATION PROCESS

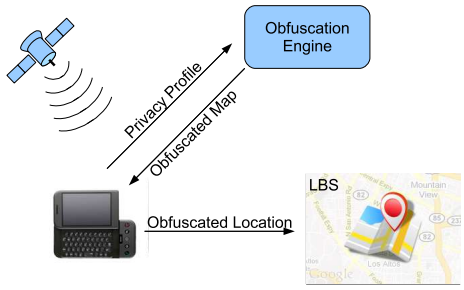


Figure 3: Reference architecture

Figure 3 shows the reference architecture of the PROBE system. PROBE assumes a conventional networked architecture consisting of a LBS server and a set of GPS-equipped mobile clients. The core component of the system is the *Obfuscation Engine*. The Obfuscation Engine computes the obfuscated locations. For the sake of efficiency, the obfuscation process is organized in two phases called *off-line* and *run-time*, respectively. In the off-line phase the user invokes the *Obfuscation Engine* to compute the *obfuscated map* based on the privacy profile. We emphasize that this way the obfuscated locations are all computed before any request is made with consequent gain of efficiency at run-time. Moreover, the obfuscated maps are to be re-calculated only when the geographical database or the user profile changes. At run time, upon a service request, the client simply matches the user's position against the obfuscated map. If the position falls inside an obfuscated location, then the actual position is replaced by the coarser position which is then transmitted to the LBS provider. Otherwise the position is transmitted without changes.

A key choice concerns the implementation of the obfuscation algorithm. Computing an obfuscated map means to determine a set of regions which satisfy the constraints specified in the privacy profile. The problem is not trivial because

an obfuscated region may have a complex shape depending on the spatial distribution of user population; moreover, it is necessary to limit the loss of geometric precision because an obfuscated position can be arbitrarily large and that may compromise the quality of location information.

4.1 Outline of the strategy

A flexible approach is to adopt a grid-based representation of space [1]. Assume space to be subdivided in cells of regular and sufficiently small size. Features are typically agglomerates of cells. Given a sensitive feature type FT_o and an instance F_o, each cell c within F_o has sensitivity $P_{sens}(FT_o, c) = 1$. Since such sensitivity is greater than the threshold sensitivity value we say that c is *over-sensitive*. The idea is to obfuscate each cell c which is over-sensitive by progressively aggregating neighbor cells. The process develops by progressively enlarging the region containing c until an obfuscated location is achieved or the region degenerates in the whole space. A key question is whether this method leads to a solution. It can be shown that, for how the sensitivity metric is defined, the method converges towards a solutions if that exists.

4.1.1 Soundness of the aggregation method

We introduce first some preliminary definitions. Let $GR = \{c_1, c_2, ...c_n\}$ be the grid defined over the reference space and $\mathcal{C}$ be a partition (not necessarily the initial one) of GR. Two cells $c_1, c_2 \in \mathcal{C}$ are adjacent if they have a common border. Given two adjacent cells c_1, c_2, the operation which merges the two cells generates a new partition $\mathcal{C}'$ in which cells c_1 and c_2 are replaced by cell $c = c_1 \bigcup^S c_2$. We say that partition $\mathcal{C}'$ *is derived* from partition $\mathcal{C}$, written as $\mathcal{C}' \succcurlyeq \mathcal{C}$. Consider the set $\mathcal{P}_{\mathcal{C}_{in}}$ of partitions derived directly or indirectly from the initial partition $\mathcal{C}_{in}$ through subsequent operations of merge. The poset $H = (\mathcal{P}_{\mathcal{C}_{in}}, \succcurlyeq)$ is a bounded lattice in which the least element is the initial partition while the greatest element is the partition consisting of a unique element, that is, the whole space (called *maximal partition*).

We now show that when two cells are merged, the sensitivity of the resulting cell is lower than the maximum between the sensitivity values of the two starting cells. Then it is shown that the maximum sensitivity value in a partition $\mathcal{C}_A$ (wrt each feature type and each region) is weakly anti-monotonic with respect to the "is derived" relation, that is, that the maximum sensitivity of a partition is equal or greater than the maximum sensitivity of a derived partition. Finally we show that whenever the sensitivity of the reference space Ω is known, one can determine whether at least one solution exists. Proofs are reported in [2].

THEOREM 4.1. *Let c_1 and c_2 be two cells of a partition $\mathcal{C}$, and $c = c_1 \cup^s c_2$. Then, the region resulting from the merge of two cells has a sensitivity which is not higher than the maximum sensitivity of the initial cells, that is:*

$$P_{sens}(ft, c) \leq \max\{P_{sens}(ft, c_1), P_{sens}(ft, c_2)\}$$

THEOREM 4.2. *Let $\mathcal{C}_A$ and $\mathcal{C}_B$ be two distinct partitions of $P_{\mathcal{C}_{in}}$ and let $ft \in FT_S$ be a sensitive feature type. Then the maximum sensitivity of a partition is equal or greater than the maximum sensitivity of a derived partition, that is:*

$$\mathcal{C}_A \succcurlyeq \mathcal{C}_B \Longrightarrow \max_{r \in \mathcal{C}_A} P_{sens}(ft, r) \leq \max_{r \in \mathcal{C}_B} P_{sens}(ft, r)$$

THEOREM 4.3. *Necessary condition for an obfuscated map to exist is that the whole reference space Ω satisfies the user's privacy preferences.*

4.2 The obfuscation algorithm

We emphasize that the above aggregation method is applied to the single cells and not to the whole sensitive feature. A single feature can thus be obfuscated by several regions, each covering a portion of such feature. An advantage of this method is that by fragmenting the original sensitive region and expanding each fragment separately, one can generate obfuscated locations which are smaller than the regions that would be obtained by obfuscating the entire region. The result is a finer-grained obfuscation and thus a better QoS.

We now present an obfuscation algorithm, called $Sens_{Hil}$, which applies the above strategy to provide fine-grained obfuscated locations. $Sens_{Hil}$ maps the grid onto a Hilbert space-filling curve and then performs cell aggregation over such a space. The Hilbert space-filling curve is a one-dimensional curve which visits every point within a discrete two-dimensional space. Similarly to the approaches in [10, 9], we exploit the locality property of Hilbert curves [17] to generate obfuscated locations. In our algorithm, a cell c is obfuscated by progressively aggregating the cells which are close to c in the linear ordering. An obfuscated location is thus simply defined by an interval in the linear space.

Algorithm 1 $Sens_{Hil}$ Algorithm

```
                                  ▷ Obfuscate grid using privacy profile pp
 1: function HILOBFUSCATE(grid, pp)
 2:     S ← ∅                                          ▷ Obfuscated regions
 3:     for idx ← 0 ... maxHilbertIdx(grid) do        ▷ Hilbert scan
 4:         cell ← getHilbertCell(idx)
 5:                                  ▷ Get the current (one-cell) interval
 6:         if OverSensitive(r, pp) then
 7:             r ← GENERALIZEFORWARD(cell, grid, pp)
 8:             add(S, r)
 9:             idx ← r.last
10:         else if Sensitive(r, pp) then
11:             add(S, cell)
12:         end if
13:     end for
14:     FIXBACKWARD(S,pp)          ▷ Fix the last interval if needed
15:     return S
16: end function
```

4.2.1 Details of the algorithm $Sens_{Hil}$

Algorithm 1 details the function *HilObfuscate* generating an obfuscated map. The algorithm consists of two phases. The first phase is called *forward generalization*. The algorithm starts scanning the cells sequence (in the linear ordering) from the first cell.

As an over-sensitive cell is found, the algorithm attempts to generate a obfuscated interval r starting from *cell* (function GENERALIZEFORWARD in Algorithm 2). If such interval is found, r is inserted into the result set S and the scan proceeds until every cell has been examined. In case *cell* is sensitive, but not oversensitive, no further generalization is needed and the one-cell interval representing the cell is inserted into S.

Upon completion of the scan, it may happen that the last sensitive cell cannot be generalized, because, for example, represents the last cell in the cell sequence. If this is the case, the algorithm expands the current interval back-

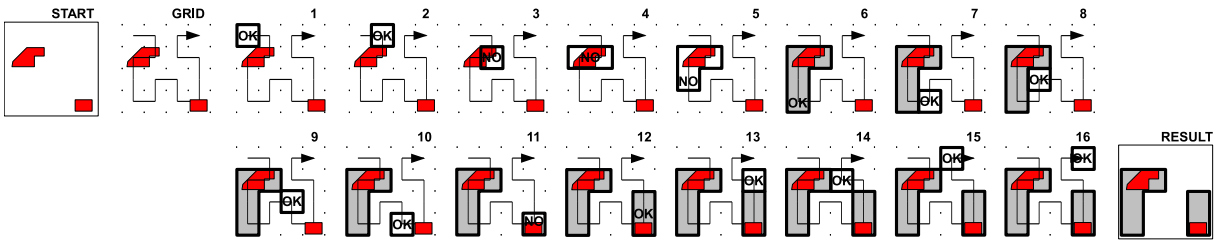


Figure 4: $Sens_{Hil}$ An example. Subfigures numbered from 1 to 16 illustrate the intermediate steps between START and RESULT.

Algorithm 2 $Sens_{Hil}$ Algorithm subroutines

```
1: function GENERALIZEFORWARD(r, grid, pp)
2:     for idx ← r.first ... maxHilbertIdx(grid) do
3:         r.last ← idx                          ▷ Expand current interval
4:         if ¬OverSensitive(r, pp) then
5:             return r
6:         end if
7:     end for
8:     return r
9: end function

        ▷ Backward expansion if the last interval r violates pp
10: procedure FIXBACKWARD(S, pp)
11:     r ← last(S)
12:     if OverSensitive(r, pp) then
13:         r ← GENERALIZEBACKWARD(r, grid, pp)
14:         if OverSensitive(r, pp) then
15:             S ← ⊥                             ▷ Obfuscation failed
16:         else
17:             add(S, r)                         ▷ Also remove redundancies
18:         end if
19:     end if
20: end procedure

21: function GENERALIZEBACKWARD(r, grid, pp)
22:     for idx ← r.first − 1 ... 0 reverse do
23:         r.first ← idx                         ▷ Expand backward
24:         if ¬OverSensitive(r, pp) then
25:             return r
26:         end if
27:     end for
28:     return r
29: end function
```

wards until a convenient interval is found or the entire sequence of cells is scanned again from the last cell to the first one (function FIXBACKWARD in Algorithm 2). This phase is called *backward generalization*. Note that in order to ensure that intervals are disjoint, the addition of r to S in the backward phase through the *add (S,r)* operation, at line 17, may entail a change of the set S. For example, the operation $add(\{[1,2],[4,7],[8,9]\},[5,12])$ results into the set $S = \{[1,2],[4,12]\}$. Each cell is examined at most twice, once per phase. The overall complexity of the $Sens_{Hil}$ algorithm is, thus, $O(|GR|)$, where $|GR|$ is the number of grid cells. It can be shown that the non-empty set $HilObfuscated(GR, pp)$ is an obfuscated map.

EXAMPLE 4. *Figure 4 illustrates the step-wise execution of the algorithm applied to a grid containing two sensitive features. The initial grid is labelled as 'START' in the figure (on the left) while the obfuscated map is labelled 'RESULT' (on the right). The two sensitive features are of the same type. Further, we set the privacy threshold to* 25%. *The reference space is described by a* 4×4 *grid, consisting of 16 cells. Cells are transversed following the Hilbert ordering (subfigure labeled as 'GRID', on the left). Subfigures 4.1–16 show the steps of the algorithm. At each step, the region being examined is labelled either 'OK' or 'NO', depending on whether the privacy preference is satisfied or not. Step 1 and 2 simply skip the first two cells in the ordering because not sensitive. The* 3^{rd} *cell, instead, is largely covered by sensitive features. Thus, in the subsequent steps (subfig. 3-6) the cell is progressively aggregated with neighbor cells until an obfuscated region is found (subfig. 6). The algorithm proceeds until the last cell has been processed.*

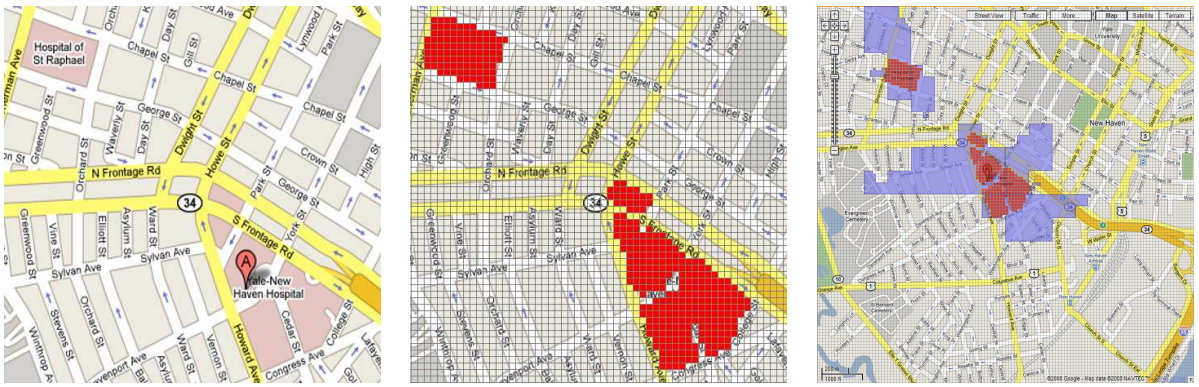
4.3 An example of obfuscation

In Figure 5 we apply $Sens_Hil$ to obfuscate two existing hospitals located in New Haven (US), named Hospital of S. Raphael and Yale-New Haven Hospital respectively. The two hospitals represent two features of the sensitive feature type Hospital. Figure 5.a. zooms on the two sensitive locations, situated respectively on the North and on the South. Figure 5.b shows the cell-based representation of the hospitals extent in a grid of 128×128 cells. Cells have a size of about 20 metres. The cells covering the hospitals have sensitivity value 1 wrt the feature type Hospital. We generate the obfuscated map using the profile $< FT_S, T >$ with $FT_S = \{Hospital\}$ and $T = \{\mathcal{T}(Hospital) = 0.3\}$. The resulting obfuscated locations are displayed in Figure 5.c in light grey (light blue in the color version). The obfuscated map consists of 7 obfuscated locations, covering the two hospitals. It can be noticed the irregular shape of the obfuscated regions. Each shape can however be simply described by an interval in the Hilbert space-filling curve.

5. EXPERIMENTAL EVALUATION

We have made experiments to evaluate various parameters using grids of fixed and varying size, with different percentage of sensitive cells (referred to as *coverage*) and privacy thresholds. We assume that all user's positions are equally probable, except in those regions which are explicitly defined to be unreachable. $Sens_{Hil}$ is implemented in Java, using the library [13]. The experiments were run on a laptop PC equipped with an AMD Turion Mobile MT30 1.6GHz CPU, 1,37GB of RAM and Windows XP.

We have run the experiments over synthetic data. For the generation of synthetic data we developed the *Spatially-aware Generalization* (SAG, for short) tool. SAG enables the generation of grids randomly populated by features of user-defined type. Features have a rectangular shape, of varying size, and are represented as group of cells. Each cell c is either empty or completely covered by a feature



(a) (b) (c)

Figure 5: Obfuscated map generated by the $Sens_{hil}$ algorithm for two hospitals in New Haven, Connecticut

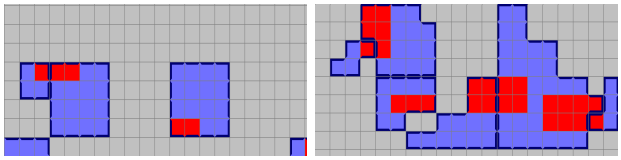


Figure 6: Obfuscated regions created by: (left) $Sens_{Pyr}$, (right) $Sens_{Hil}$

of feature type ft. Thus, $P_{sens}(ft, c) \in \{0, 1\}$. Grids are generated based on the following parameters: the size of the grid; the set FT of feature types; for each type $ft \in FT$, the percentage of cells covered by ft; the features types denoting unreachable regions, such as lake. Hence SAG populates the reachable portion of space with rectangles of varying size. The side of each rectangle is generated on a random basis using a binomial distribution. The average size of the rectangle side is 3 cells and the range is [0,6]. SAG also enables the specification of privacy profiles. The user flags the features types in the set FT that are sensitive and then specifies for each sensitive feature the threshold value.

5.1 The experiments

For comparison purposes, we developed the algorithm $Sens_{Pyr}$ [2] which provides an alternative implementation of the PROBE obfuscation strategy. $Sens_{Pyr}$ uses a pyramid data structure to represent the space grid similar to the structure used for spatial k-anonymization in the Casper system [12]. The pyramid takes the form of a tree in which the nodes represent the regions obtained by recursively subdividing space in four quadrants until the base cells are reached. The root at level 0 corresponds to the entire reference space; the leaves are the cells of the finest-grained grid; a cell c which is not a leaf has four children, one for each quadrant of the region denoted by c. The process of cell aggregation works as follows: first, each leaf which is over-sensitive is aggregated with the cells of the quadrant the cell belongs to. If the resulting region remains over-sensitive, the aggregation is possibly recursively applied to the cells of the grid at the immediately lower level in the pyramid. Figure 6 highlights the different shape of the obfuscated locations generated by the two algorithms (the red cells are sensitive, the regions including blue cells are the obfuscated locations).

We have made five experiments with the two algorithms. Experiments 1,2,3, and 5 use a grid of size 1024×1024 cells. At a resolution of 10 metres, the reference space is thus about $10km \times 10km$ which is the size of an average city. The independent variable in the experiments is the coverage which ranges in the interval $[1, 45]$, which seems a reasonable choice; a value of x means that the percentage of sensitive cells is $x\%$. Further, we consider three possible values for the threshold function, that is, $\mathcal{T}(ft) \in \{0.1, 0.2, 0.4\}$. Each algorithm is run 100 times, for different values of the coverage and the threshold value and average values are reported for each experiment. Experiment 4 evaluates the two algorithms on grids of increasing size ranging between 64×64 and 4096×4096 with fixed coverages equal to 0.5% and 10%.

Experiment 1: Success rate. The outcome is the rate of successful generation of obfuscated maps (*success rate*). As the coverage increases and the privacy requirements become more restrictive, the probability of failure in the map generation increases. The graphs in Figure 7 show that the generation is successful until the percentage of coverage is below a *breaking* value. For example, when the threshold has value 0.4, the breaking value is nearly 40. It can be noticed that the breaking values are nearly the same for the two algorithms.

Experiment 2: Average number of obfuscated regions. The outcome is the average number of obfuscated locations computed by the two algorithms when the map generation process does not fail. The two graphs in Figure 8 show that the number of obfuscated regions generated by $Sens_{Hil}$ is significantly higher than the number of obfuscated regions generated by $Sens_{Pyr}$. It can be noticed that the cardinality increases up to a maximum value and then decreases. The reason of such behavior is that for low percentages of coverage, the number of cells to obfuscate is relatively low. The number of obfuscated regions however increases up to a maximum value. When the density of sensitive cells is too high the algorithms generate large obfuscated areas and thus the number of obfuscated regions globally decreases.

Experiment 3: Average size of the obfuscated location. The outcome is the average number of cells in an obfuscated region. Not surprisingly, the graphs in Figure 9 show that the $Sens_{Hil}$ generates more precise obfuscated maps than $Sens_{Pyr}$. Quantitative values are reported later on. It can be noticed that the size of the obfuscated maps

grows very rapidly, especially for $Sens_{Pyr}$, as the percentage of coverage becomes closer to the breaking point.

GridSize	NReg	NCell	GenTime(ms)
64×64	28	112	0.49
128×128	106	120	2.08
256×256	429	118	11.5
512×512	1726	118	72
1024×1024	6943	117	185
2048×2048	27735	117	758
4096×4096	111026	117	3255

Table 1: Measures for $Sens_{Pyr}$

GridSide	NReg	NCell	GenTime(ms)
64×64	43	46	0.49
128×128	175	47	2.17
256×256	700	47	8.8
512×512	2835	46	31
1024×1024	11372	46	177
2048×2048	45483	46	543
4096×4096	181920	46	2104

Table 2: Measures for $Sens_{Hil}$

Experiment 4: Grid size. Table 1 and Table 2 report the measures obtained by running the two algorithms over grids of increasing size ranging between 64×64 and 4096×4096 with a 10% coverage. Each table row specifies: the grid size (GridSize), the average number of obfuscated regions (NReg), the average number of cells per regions (NCell), and the map generation time (GenTime). If we look at the experiments over a grid of 1024×1024 we observe that:

- The number of obfuscated regions generated by $Sens_{Hil}$ is about 40% higher than the number of regions generated by $Sens_{Pyr}$.
- The average number of cells per obfuscated regions in $Sens_{Hil}$ is 46 against 118 of $Sens_{Pyr}$. At the given resolution ($10m \times 10m$) the average area of the obfuscated region generated by $Sens_{Hil}$is thus $4600m^2$.
- The map generation time for $Sens_{Hil}$ is 177 ms against 185 ms of $Sens_{Pyr}$. Thus, the performance is thus not significantly different, but the $Sens_{Hil}$ computes significantly more precise obfuscated locations than $Sens_{Pyr}$.

The graph in Figure 10 shows that the generalization time increases linearly with the size of the grid for both algorithms. Moreover, such a time is not significantly affected by the coverage for coverages that are not close to the breaking point. Note that the generation time is high (few seconds) when the grid is 4096×4096. Consider, however, that these experiments have been run on a laptop.

Experiment 5: Obfuscation ratio. The outcome is the *obfuscation ratio* of an obfuscated map, that is the ratio of the total number of cells contained in obfuscated locations over the total number of sensitive cells. Figure 11 reports the result of the experiment for three different privacy thresholds and a varying average coverage value. For example, for a privacy threshold equal to 0.1 and coverage equal to 2%, $Sens_{Pyr}$ generates an average of about 18 obfuscated cells for each sensitive cell, whereas $Sens_{Hil}$ presents an obfuscation ratio which is about 10. It is important to observe

Grid size	# Obfuscated locations	$\approx Size(KB)$
512×512	2835	22
1024×1024	11372	90
2048×2048	45483	363
4096×4096	181920	1455

Table 3: Avg. size of the obfuscated maps

that the obfuscation ratio of the maps generated $Sens_{Hil}$ is almost constant and independent of the average coverage. Notably, such obfuscation ratio is almost always equal to the best attainable one, i.e., no algorithm can obfuscate the same sensitive regions and obtain a smaller obfuscated area.

6. RUN-TIME PRIVACY ENFORCEMENT

We consider now the size of the obfuscated map sent to the client. We recall that the client requests an obfuscated map by forwarding a request to the Obfuscation Engine. Such a request contains the privacy profile. The privacy profile is a set $P = \{t_1...t_n\}$ of pairs representing the privacy preferences, with $t_i \equiv < ft, v >$, $i \in \{1..n\}$. The user can select the sensitive feature types for example from a predefined list. In order to limit the size of the obfuscated map, the user can specify the bounding box of the region of interest. The Obfuscation Engine generates an obfuscated map, if it exists. Such map consists of a non-empty set of obfuscated regions, where each region is represented by an interval $[a, b]$ with a and b Hilbert indexes. The obfuscated map is then stored on the client as a B-tree.

Regarding the size of map being generated, if the encoding of the interval representing an obfuscated region requires 8 bytes, the size of an obfuscated map is $n \times 8$ bytes, where n is the number of regions in the obfuscated map. Based on the experiments reported the previous section, Table 3 reports the average size of the obfuscated maps generated for grids of varying size, assuming a 10% coverage.

6.1 Privacy Enforcement

At run time, the client computes the location information to be transferred to the LBS provider. This operation is referred to as *privacy enforcement*. The correspondent algorithm is reported in Algorithm 3. Given user's position p, and obfuscated map S, the algorithm maps p onto a Hilbert index and then checks whether such a value is included in an interval of the indexed obfuscated map(line 4). Because the obfuscated map can be stored as a B-tree, the complexity of the operation is $O(log\ n)$. If p does not fall inside any interval, then the function simply returns p otherwise the enclosing interval r. The result is then transferred to the LBS provider possibly along with the parameters of the Hilbert-space filling curve.

7. CONCLUSIONS

PROBE is a comprehensive system for the protection of location privacy against location inference attacks in LBS. A key feature of the system is that it allows the subscribers of a LBS to specify location privacy preferences about the places that they consider sensitive as well as the desired degree of privacy protection. As part of PROBE we have also developed a technique for efficiently computing obfuscated maps that are personalized based on the user privacy

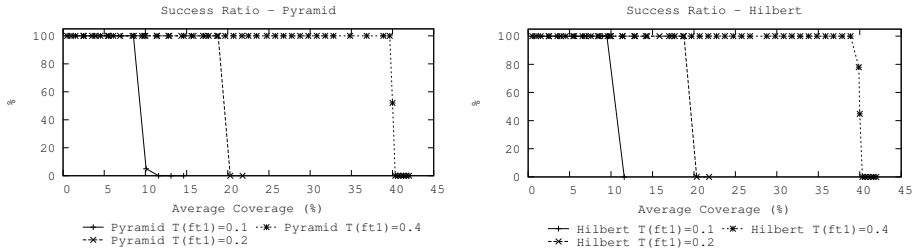


Figure 7: Success rate

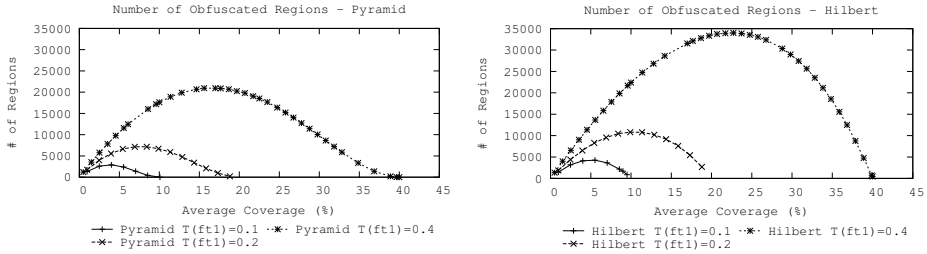


Figure 8: Avg. number of obfuscated regions

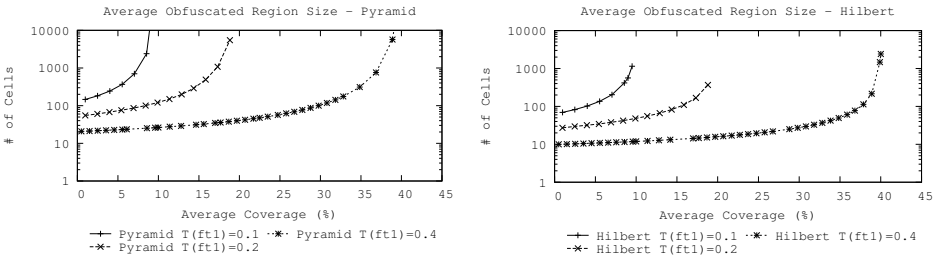


Figure 9: Avg. number of cells per obfuscated region

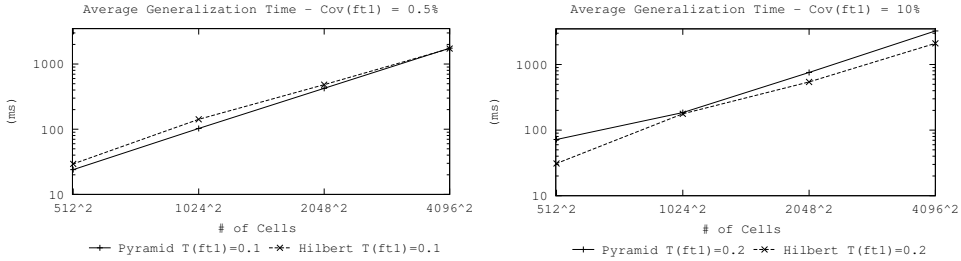


Figure 10: Avg. time for varying-size grids and different coverages.

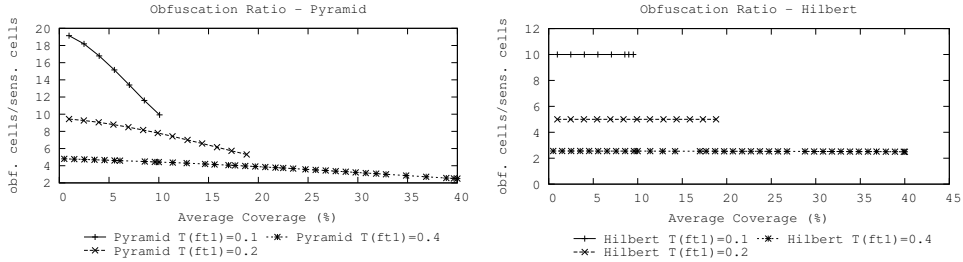


Figure 11: Avg. number of obfuscated cells per sensitive cell

Algorithm 3 Privacy Enforcer Algorithm

```
1: function PrivacyEnforcment(p, MBB, Dim, S)
2:     CellCoord ← GetCellCoord(p, MBB, Dim)
3:         ▷ Returns the row and column of the cell containing p
4:     idx ← GetHilbertIndex(CellCoord, Dim)
5:                 ▷ Returns the Hilbert index of the previous cell
6:     r = GetInterval(S, idx)
7:         ▷ Returns the obfuscated region containing idx or null
8:     if r = ⊥ then
9:         return p                  ▷ Returns the original position
10:    else
11:        return r                  ▷ Returns the obfuscated location
12:    end if
13: end function
```

preferences. The technique has very small storage requirements and thus it is suited for use on small devices, such as cellular phones. We emphasize that the PROBE method is not antagonist to spatial k-anonymity but rather complementary. Further, PROBE can be used in a wide range of novel and challenging applications, for example to protect the privacy of sensitive locations in geo-social networks applications. Our work leaves room for diverse research directions. In particular we highlight three major issues: a) the protection of sensitive locations along user's trajectories. b) To prevent inferences on the user's profile. The privacy profile can be itself sensitive. For example, if Bob detects that night clubs are sensitive locations for Alice, then Bob can also infer that Alice is used to go to night clubs. c) To improve the obfuscation algorithms.

8. REFERENCES

[1] M. Damiani, E. Bertino, and C. Silvestri. Protecting location privacy through semantics-aware obfuscation techniques. In *Proc. of IFIPTM 2008*, pages 231–245. Springer Boston, June 18-20 2008.

[2] M. L. Damiani, E. Bertino, and C. Silvestri. PROBE: an obfuscation system for the protection of sensitive location information in lbs. CERIAS Technical Report, Purdue University, 2008.

[3] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Pervasive Computing*. Springer, 2005.

[4] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *Proc. of the 25th IEEE ICDCS*, 2005.

[5] G. Ghinita, M. Damiani, E. Bertino, and C. Silvestri. Interactive Location Cloaking with the PROBE Obfuscator. In *Proc. of the Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, 2009.

[6] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L.Tan. Private Queries in Location Based Services: Anonymizers are not Necessary. In *Proc. ACM SIGMOD Conference*, 2008.

[7] 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*. ACM Press, 2003.

[8] U. Hengartner and P. Steenkiste. Access control to people location information. *ACM Trans. Inf. Syst. Secur.*, 8(4):424–456, 2005.

[9] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE TKDE*, 2007.

[10] B. Krishnamachari, G. Ghinita, and P. Kalnis. Privacy-Preserving Publication of User Locations in the Proximity of Sensitive Sites. In *Proc. SSDBM*, 2008.

[11] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-Diversity: Privacy Beyond k-Anonymity. In *Proc. ICDE*, 2006.

[12] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new Casper: query processing for location services without compromising privacy. In *Proc. VLDB*, pages 763–774, 2006.

[13] D. Moore. C Library Hilbert.c. http://www.caam.rice.edu/ dougm.

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

[15] Open GIS Consortium. Open GIS simple features specification for SQL, 1999. Revision 1.1.

[16] N. Poolsappasit and I. Ray. Towards Achieving Personalized Privacy for Location-Based Services. *Transactions on Data Privacy*, 2:1:77 – 99, 2009.

[17] H. Samet. *Foundations of Multidimensional and Metric data Structures*. Morgan Kaufmann, 2006.

[18] E. Snekkenes. Concepts for personal location privacy policies. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 48–57, New York, NY, USA, 2001. ACM Press.

[19] X. Xiao and Y. Tao. Personalized privacy preservation. In *Proc. of the 2006 ACM SIGMOD*, pages 229–240, New York, NY, USA, 2006. ACM.

[20] P. H. Xue M., Kalnis P. Location Diversity: Enhanced Privacy Protection in Location Based Services. In *Proc. of the International Symposium on Location and Context Awareness (LoCA)*, 2009.

[21] M. L. Yiu, C. Jensen, X. Huang, and H. Lu. SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services. In *proc. IEEE 24th International Conference on Data Engineering*, 2008.

[22] M. Youssef, V. Atluri, and N. R. Adam. Preserving mobile customer privacy: an access control system for moving objects and customer profiles. In *Proc. MDM*, 2005.

Location-aware Privacy and More: A Systems Approach using Context-aware Database Management Systems*

Walid G. Aref
Department of Computer Science
Purdue University
West Lafayette, Indiana 47907
aref@cs.purdue.edu

Hicham G. Elmongui
Department of Computer Science
Purdue University
West Lafayette, Indiana 47907
elmongui@cs.purdue.edu

Mourad Ouzzani
Cyber Center
Purdue University
West Lafayette, Indiana 47907
mourad@cs.purdue.edu

ABSTRACT

When a user issues a query, database engines will usually return results based solely on the query and the content of the database. However, query issuers have a "context" which if taken into account will certainly change the outcome of the query. Thus, when responding to the query, the database system can consider the query issuer's context and return only the objects/tuples in the database that not only satisfy the query predicates but also are relevant to the query issuer's context. In this paper, we give an overview of Chameleon; a context-aware database management system. Chameleon introduces SQL-level constructs that describe the "context" in which the query is issued as well as the reciprocal contexts of the objects in the database. By tying the query issuer's contexts with the corresponding contexts of the objects in the database, Chameleon can retrieve the objects of relevance to the query context. Using Chameleon's general interfaces for context definition and awareness activation, we show how database systems that offer not only location-sensitive privacy but also other forms of privacy, e.g., both location-sensitive and time-sensitive privacy profiles for their users can be realized easily. Several modeling and performance challenges for realizing context-aware database management systems are presented.

Categories and Subject Descriptors

H.2 [**Database Management**]. H.2.1 [**Logical Design**]: Data models.

General Terms

Algorithms, Management, Performance, Design, Languages.

Keywords

Context awareness, privacy, preferences, personalization, database systems

1. INTRODUCTION

As applications and application requirements grow in complexity, the underlying data management system has to increase in sophistication to cope with this complexity. In the early days, when applications, e.g., Geographic Information Systems (GIS), demanded efficient handling of large amounts of spatial data, DBMSs had to increase in sophistication to handle location data efficiently. This included the costly development of spatial indexing techniques that support concurrency and recovery, spatial query processing algorithms, e.g., spatial join algorithms with and without spatial indexes, and locality preserving strategies for disk placement of spatial data.

Similarly, Hippocratic databases have been proposed to address the privacy policy requirements that users' data are being used only by the intended recipients and only for the purposes approved by the data owners [1]. Systems, e.g., Hippocratic PostgreSQL [2] have been prototyped to provide controlled disclosure of the users' data according to the users' approved privacy policies.

In order for database systems to provide users with location-aware privacy, tremendous effort has to take place to develop combined Hippocratic and spatial database engines, which is very costly. In this paper, we present a systems approach to address this issue. Chameleon, a context-aware DBMS, is an extensible database server that uses contexts to eliminate the need for tailoring specialized engines [3], e.g., a spatial database engine, a Hippocratic database engine, a location-sensitive Hippocratic database engine, a time-sensitive, location-sensitive Hippocratic database engine (refer to Figure 1). Instead, using Chameleon, one can realize these systems by defining appropriate contexts using Chameleon's context definition and manipulation languages.

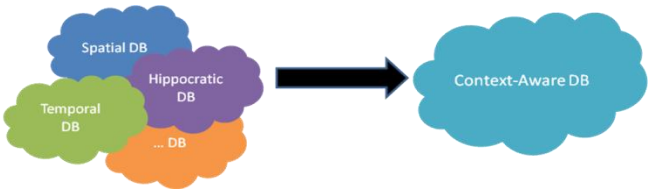


Figure 1: The vision behind Chameleon.

Chameleon supports two notions of context: the context surrounding the query issuer and the reciprocal contexts of the objects stored in the database. The query context reflects the situation of the query issuer, e.g., the query issuer's location, the time the query is issued, the identity of the query issuer, or even the temperature or the weather conditions surrounding the query issuer. Chameleon takes these situations into consideration when answering a query. For example, in Figure 2, when querying the database asking for a close-by restaurant, the user wants the database system to return restaurant responses that match the user's current context, i.e., her location, the time the query is issued, and her personal diet and dietary restrictions.

*The authors acknowledge the support of the National Science Foundation under Grant IIS-0811954.

ACM SPRINGL '09 November 3, 2009. Seattle, WA, USA

Database objects have contexts that are reciprocal to the query issuer's contexts, e.g., (refer to Figure 3), the location of the database objects, the time duration of an object (or when the object can be available for querying), and the identity of the object (or the ids of the query issuers or classes of query issuers that are allowed to access the object).

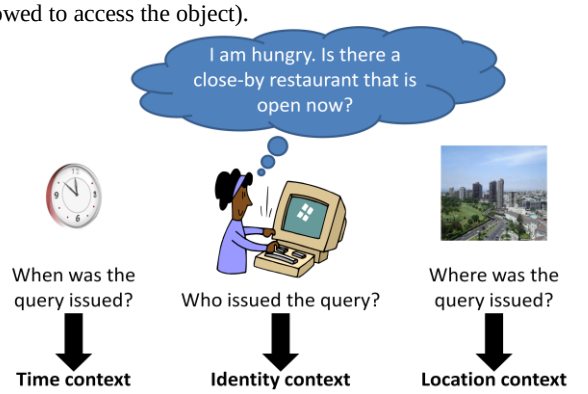


Figure 2: Chameleon considers the user's active contexts (user's time, location, and identity) when responding to the user's query.

In Chameleon, we can combine multiple contexts into more complex ones using the proposed context composition, e.g., a Hippocratic DBMS that also is location- and time-sensitive by combing the location-, temporal-, and identity-sensitive contextual services.

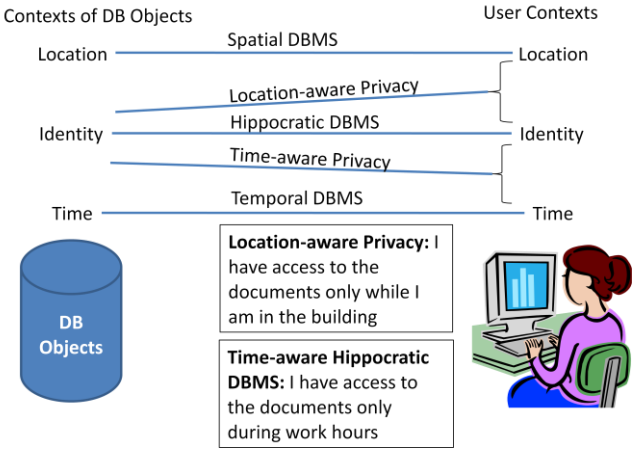


Figure 3: Illustration of what user and DB object contexts are combined in Chameleon to realize various specialized DB engines using the same context interfaces in Chameleon.

In this paper, we give several proof-of-concept instantiations of Chameleon, e.g., one to realize a privacy-aware (Hippocratic) database server, and another to realize a spatial database server using the same proposed constructs and interfaces of Chameleon. Further, we show how contexts can be combined within Chameleon to realize more complex systems, e.g., a server that supports location- and time-aware privacy database, i.e., one where the privacy profiles of database objects depend not only on the identity and purpose of the query issuers but also on the query issuer's location and time when they issue the query (location-aware and time-aware privacy).

Chameleon is built using extensions to PostgreSQL that include:

(1) New syntax and query rewrite components to define contexts and to issue queries that use contexts,
(2) New query operators that process contexts, e.g., the Skyline join and FilterMark operators that are vital when processing queries that involve contexts, and
(3) Extensions to the query optimizer to invoke these new operators when appropriate.

The rest of this paper proceeds as follows. In Section 2, we present the context classes in Chameleon and the dimensions that we use to define a context. Section 3 presents the syntax and semantics of the extended SQL constructs within Chameleon that defines contexts. Section 4 addresses conceptual evaluation of context-aware SQL commands and implementation issues. Section 5 gives several example instantiations to realize a spatial database server, a Hippocratic database server, and a location-aware and time-aware Hippocratic database server. The latter illustrates how contexts can be combined in Chameleon to realize more complex servers. Section 6 discusses related work. Section 7 includes some research challenges and concluding remarks.

2. CONTEXT CLASSES

For illustration, we use a simple bookstore example, where users express their preferences when accessing the bookstore database. Later in the paper, we will show more sophisticated cases, mainly for system realizations of location-aware privacy as well as location- and time-aware privacy.

Table 1 gives a projection on the table books that contains information about books in a certain bookstore. Among other pieces of information included in this table, we can find the name of a book, the years of its publishing, the category under which this book falls, the type of the cover (HC for hardcover or PB for paperback), as well as whether or not the book is in stock. Only the books in stock that are relevant to the user's context are retrieved.

In contrast to existing work on context-aware systems that are built on top of a database, we propose to incorporate context awareness inside the DBMS. We adopt a broad definition of what a context is. For example, the physical location in space of the query issuer when he/she issues the query can be part of the query context. The time the query is issued and the identity of who issued the query may also both be part of the query context. We support the following two classes of contexts:

(1) The User Context, i.e., the context of the query issuer.

(2) The Object Context, i.e., the contexts of the queried data.

Figure 4 illustrates a high-level view of Chameleon's context model. We classify user contexts according to three dimensions. These dimensions will be used when the application developer defines a context in Chameleon. These dimensions will reflect in the access method selection of any query on the tables that are affected by that context.

Dimension 1 - Context Sign: The sign of a user context is either "positive" (S) or "negative" (G). A positive context defines what the context is. For instance, if the context is location, an instance of a positive context is the preferred locations by the user, e.g.,

specified as a range. On the other hand, a negative context defines what the context is not. An instance of a negative location context

row	title	year	category	cover	instock
1	book01	2004	Science Fiction	HC	√
2	book02	2002	Travel	PB	√
3	book03	2001	Medicine	HC	X
4	book04	2000	Cooking	PB	√
5	book05	1997	Science Fiction	PB	√
6	book06	2001	Medicine	HC	√
7	book07	1995	Cooking	PB	X
8	book08	1996	Travel	PB	√
9	book09	2000	Science Fiction	PB	√
10	book10	2003	Medicine	PB	√
11	book11	2005	Travel	HC	√
12	book12	2006	Cooking	HC	X
13	book13	2004	Medicine	PB	√
14	book14	2006	Science Fiction	HC	√
15	book15	2005	Travel	HC	√
16	book16	2006	Cooking	HC	√
17	book17	1976	Medicine	PB	√
18	book18	2001	Travel	PB	√
19	book19	2007	Science Fiction	HC	√
20	book20	1988	Cooking	PB	X
21	book21	1993	Science Fiction	PB	√
22	book22	2006	Medicine	HC	X
23	book23	1999	Cooking	PB	X
24	book24	2006	Medicine	HC	√
25	book25	2006	Travel	PB	√

Table 1: The example bookstore database.

is the locations or regions not desired by or prohibited to the user. In the running example, an instance of a positive context is the willingness to buy hardcover books only. However, trying to avoid science fiction books is a case for a negative context.

Dimension 2 - Contextual Relation: The contextual relation is the relation among the contextual data. This relation mainly shows the order of relevance of the contextual data. The contextual relation can be an equivalence relation (Q). In this case, data that comply with all contextual values are reported with no special ordering. Besides, the contextual relation can also be a total ordering relation (T). This relation would reflect on the data being reported to the user. The data will be sorted on the rank of the contextual values with which the data conform. Moreover, the contextual relation can be a partial ordering relation (P). In contrast to the previous relation, the rank of the contextual values here will follow a partial order rather than a linear order. Referring to the books table, an example of an equivalence relation is the equal willingness to buy a science fiction book or a travel book. However, if the user is interested in new books, a total ordering relation would be more appropriate to retrieve the latest books first. If the user prefers cooking books over science fiction books, and travel books over medicine books, with no specific preference among the other combinations, would need to specify her context to contain partially ordered contextual values. Partially ordered values may be transformed into linear ordered values using an appropriate (possibly online) topological sort algorithm. This is out of the scope of this work, but we add the modeling part here for completeness.

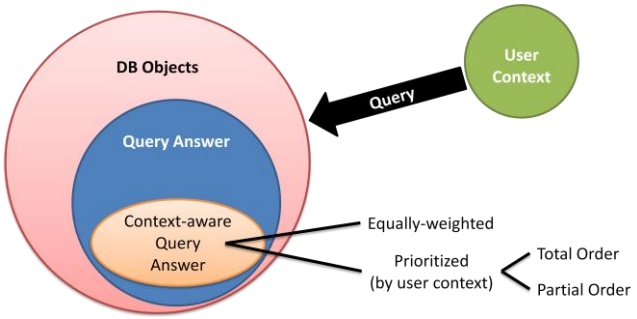


Figure 4: Abstraction of Contexts in Chameleon.

Dimension 3 - Listing of Data: By listing of data we refer to how the data should be listed. Specifically, should the data that does not conform to the user context be excluded from the listed data? Or, should those data be included but come last? The former case is termed "unlisted excluded" (X), whereas the latter is termed "unlisted included" (N).

Consider the bookstore example, if the user context is the willingness to buy travel books only, the user context gets the "unlisted (other book categories) excluded". Nevertheless, an "unlisted included" context can be illustrated by the preference to buy hardcover books but still get the paperback books down in the list -- after retrieving all hardcover books). In a location context example, if the user context is the willingness to buy houses that lie within a certain geometric region, say R, then "unlisted excluded" means that houses outside R are not reported to the user, whereas "unlisted included" lists the houses outside R after listing the houses inside R.

2.1 User Context as a 3D Point

Based on these three dimensions, each user context is viewed as a point in the 3D space defined above. For instance, in the bookstore example, one might be willing to buy only science fiction or travel books with no particular preference between these two types. This is an example of a positive user context having an equivalence contextual relation with the unlisted contextual values excluded. Whenever a user with the aforementioned context selects all tuples from the table books, only rows 1, 2, 5, 8, 9, 11, 14, 15, 18, 19, 21, and 25 are retrieved. If the user defines the same context to have a total ordering relation instead of an

equivalence relation such that science fiction books have higher rank than travel books, the retrieved rows will be: 1, 5, 9, 14, 19, 21, 2, 8, 11, 15, 18, and 25.

All points in this 3D space are valid when the user context is positive. However, when the user context is negative, only contextual values with the unlisted included are valid. This restriction is due to the definition of a negative user context; the user is specifying what context values are not current, and hence all the others should be current (or nothing will be ever returned). Moreover, for a negative user context, since the user only describes the complement of her positive context, no rank is explicitly specified for that actual positive context. Therefore, the equivalence relation would be implicitly understood for the contextual values. Figure 5 summarizes the overall model for contexts in Chameleon along with the three dimensions. The next section illustrates how these context dimensions can be used to specify contexts using newly proposed SQL constructs.

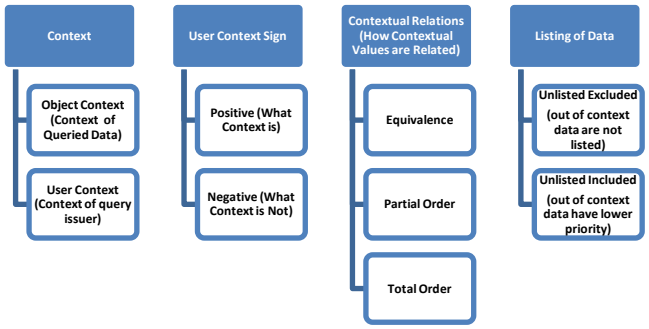


Figure 5: Conceptual Model of Contexts in Chameleon.

3. SQL EXTENSIONS FOR CONTEXT AWARENESS

In this section, we cover the various constructs that Chameleon uses to enable context awareness inside a DBMS. A brief overview of these constructs is also presented in [3].

Creating Object Contexts: Chameleon uses the CREATE OBJECT CONTEXT statement to define an object context. When the object context is part of the object relation, it does not need to be defined explicitly.

```
CREATE OBJECT CONTEXT context_name (
  {col_spec | table_constraint} [, . . . ]
  , table_binding );
```

Contextual values will be stored inside relations to be easily incorporated within the query processor. The CREATE OBJECT CONTEXT statement has similar constructs to those in the CREATE TABLE statement and it also creates an object context relation. For instance, *col_spec* refers to the specification of a column such as name, data type, default values, and so on. On a similar vein, *table_constraint* refers to any constraints on the whole context table such as check constraints.

The construct *table_binding* is the main construct that connects the object with its context. Specifically, *table_binding* has the format below.

```
BINDING KEY ([col_name [, . . . ]])
  REFERENCES ref_table [( ref_col [, . . . ] )]
  WITH bool_expr
```

The first part of the BINDING KEY is similar to the FOREIGN KEY. There are three main differences between these two types of keys. The first difference is that a foreign key in a table has to refer to a primary key in another table. This constraint does not exist for the binding key. A binding key binds the contextual value to possibly more than one object, since more than one object may exist in the same context. The second difference is that the decision to bind a contextual value with an object does not have to be equality with a column value in the referenced table. The WITH construct defines a Boolean expression that serves as the binder in case the expression evaluates to true. The third difference is that the binding key might not contain any context attribute referencing an attribute in the base table, but rather only the Boolean expression that might also contain attributes from any object context to the referenced table. Examples will be shown in the case studies section to illustrate these differences further.

Creating User Contexts: Similar to object contexts, each user context will materialize to a relation. Chameleon uses the following syntax to define a user context.

```
CREATE  [context_sign] CONTEXT context_name (
            {col_spec | table_constraint} [, . . . ]
            , table_binding [, . . . ]
            [, substitution_ key [, . . . ] ]
) [AS contextual_relation]
[WITH UNLISTED unlisted_status];

context_sign: positive | negative

contextual_relation: equivalence
                   | total order [USING ordering_func]
                   | partial order

unlisted_status: excluded | included
```

For each table affected by a user context, a binding key is used to show how the context reflects on the table. Therefore, there might be more than one binding key in a user context. Upon the creation of a user context, an implicit column is created to hold the user name of the current user. Therefore, each contextual value is associated with a certain user. Also, if an ordering relation is used for the contextual relation, then another implicit column is created to hold the rank of that contextual value. This rank can either be input by the application while acquiring contextual data, or can be computed using an ordering function ordering_func. In the latter case, the rank column does not need to exist.

Chameleon builds default indexes for context relations. Object contexts get non-clustered indexes on the context keys. User contexts are clustered in a B-tree index using the clustering key (*user_name, context_key*) if the contextual relation is equivalence. If the contextual relation is a total ordering relation, then the user context is clustered on (user name, context key) if the unlisted are to be included and on (*user_name, rank*) if the unlisted are to be excluded.

The substituting key will be discussed in detail in the next section. Populating the contextual relations will be made using standard SQL INSERT statements. Also, other data manipulation statements will still work on the contextual relations.

Global Substitution Construct: Some attributes need to be modified for presentation purposes if we want to enable context awareness. For instance, if the context is the location of a user, and the user is currently in France, then we might want all prices, in all tables, to be converted to Euro. This conversion is called global substitution, since the substitution occurs for all tables according to the current context. The substituting key defines such conversion, and is specified while defining the user context as follows.

SUBSTITUTE table_name (col_name)
BY *expression*;

The expression that substitutes the attribute can be any expression in which attributes from table_name, its object contexts, as well as the user context may appear. The substitute clause is useful in limiting the disclosure of an attribute value if the query issuer is not allowed to view that value. The substitute expression would be to display a null value instead of the original attribute value.

Setting Active Contexts: The application user may have many contexts, not all of them need to be current all the time. Therefore, we introduce the construct SET ACTIVE CONTEXT to define the current contexts to be taken into account for that user. The user_name has the CURRENT USER as a default.

SET ACTIVE CONTEXT [FOR USER user_name]
AS context_name [, . . .];
{ [WITH RANKING ORDER context_name [, . . .]]
| [WITH RANKING EXPRESSION *expression*
| [WITH SKYLINE OF *expression* {MAX | MIN} [, . . .]]};

Before querying, the command SET ACTIVE CONTEXT is issued. This command specifies the contexts to be considered when evaluating the query. It also specifies how to prioritize and combine multiple contexts.

The SET ACTIVE CONTEXT statement allows composing complex contexts from basic ones. If all the basic contexts that are used to compose a complex context have equivalence contextual relations only, then the order of executing the contexts is given by the order the contexts are listed in the AS clause.

In Chameleon, we support three different ways for combining contexts (refer to Figure 6). Consider the following two contexts C1 and C2 that are set to be active upon issuing a query. We use two contexts for simplicity in the presentation.

- Ordering:
  WITH RANKING ORDER Context1, Context2
- Multi-feature Ranking:
  WITH RANKING EXPRESSION
  2*Context1.rank+Context2.rank
- Skyline:
  SKYLINE Context1 MIN, Context2 MAX

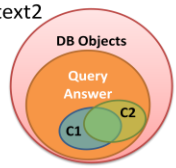


Figure 6: Examples of Three Ways for Combining Contexts in Chameleon.

Option 1 - Ordering: Using this option, Chameleon can combine the contexts by simply listing the contexts in some order, e.g., C1, C2. In this case, objects in the database are ordered according to Context C1 with ties broken according to the order within C2.

Option 2 - Multi-feature Ranking: We can combine the contexts C1 and C2 according to a weighted ranking function of C1's and C2's individual ranks.

Option 3 - Skyline Ordering: Skyline ordering is needed when the multiple contexts are independent and their ranks cannot be aggregated together. We can combine the individual contexts by returning the tuples that are not dominated by any other tuples (see Figure 7). The skyline operator [18] is used for that purpose. The WITH SKYLINE clause is used to specify to the skyline operation which expressions to use as the input ranks in the computation.

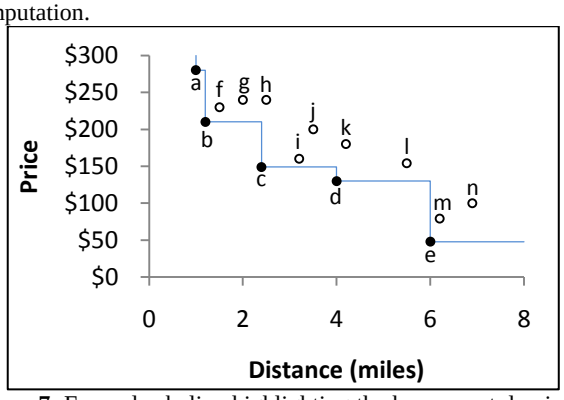


Figure 7: Example skyline highlighting the houses not dominated by other houses with respect to price and closeness to the beach.

Querying given the Active Contexts: When issuing a query, the query issuer invokes the active contexts in the following way:

WITHIN_MY_CONTEXT <Select Statement>

This command invokes the active contexts set by the SET ACTIVE CONTEXT command when evaluating the query.

4. CONCEPTUAL EVALUATION

In this section, we show why the above constructs enable context-aware query processing. We continue with our running example where someone is accessing the database of a bookstore. Only the books in stock that are relevant to the user's context are retrieved. Examples of contexts are given, their definitions using the above constructs are provided, and then we show how they are evaluated to give the desired results. First, we start by simple contexts, and later we show how these contexts are combined together to compose more complex contexts. In all the scenarios below, the user is executing the following query, and the results are the relevant tuples.

```
SELECT *
FROM books
WHERE books.instock;
```

Context 1: *The user has a preference for only books of a certain category (e.g., Science fiction).*

This context may be defined as:

```
CREATE POSITIVE CONTEXT ctxt_category_SQX (
    category varchar(20),
    BINDING KEY (category)
        REFERENCES books(category)
) AS EQUIVALENCE WITH UNLISTED EXCLUDED;

SET ACTIVE CONTEXT AS ctxt_category_SQX;
```

We give the suffix SQX to the context name above to emphasize that it is a positive [S] context with an equivalence [Q] contextual relation and that the unlisted categories in the context are to be excluded [X]. For the above example, when the user issues Q_u above, the actual query that is executed is given below. Typically, the binding key is used to join the books table with the context table, and only the books whose category exists in the context are to be returned. The following query reflects this semantic.

```
SELECT T.*
FROM books T
   INNER JOIN ctx_category_SQX C1
      ON(T.category = C1.category
      AND C1.user_name = CURRENT_USER)
WHERE T.instock;
```

Context 2: *The user's preference is for books published in 2005, and then those published in 2006 before all other books.*

This context may be defined as:

```
CREATE POSITIVE CONTEXT ctxt_year_STI (
    year integer,
   BINDING KEY (year) REFERENCES books(year)
) AS TOTAL ORDER WITH UNLISTED INCLUDED;

SET ACTIVE CONTEXT AS ctxt year STI;
```

Again, the suffix STI of the current context emphasizes that it is a positive [S] context with a total order [T] contextual relation and that the unlisted years in the context are to be included [I]. For the above example, in response to Q_u, the actual query that is executed is given below. Typically, the binding key is used to join the books table with the context table. In this case, the type of join is a left outer join, and therefore, all books will be returned at the end. The output rows are to be sorted based on the year rank, which is specified implicitly in the context as it is an ordering context. Rows with NULL context rank appear later in the list. The following query reflects this semantics.

```
SELECT T.*
FROM books T
   LEFT OUTER JOIN ctx_year_STI C1
      ON(T.year = C1.year
      AND C1.user_name = CURRENT_USER)
WHERE T.instock
ORDER BY C1.rank;
```

Context 3: *The user prefers hardcover over paperback books.*

This context may be defined as:

```
CREATE POSITIVE CONTEXT ctxt_cover_STX (
    cover integer,
   BINDING KEY (cover) REFERENCES books(cover)
) AS TOTAL ORDER WITH UNLISTED EXCLUDED;

SET ACTIVE CONTEXT AS ctxt_cover_STX;
```

For the above example, in response to Q_u, the actual query that is executed is given below. Typically, the binding key is used to join the books table with the context table. The output rows are to be sorted based on the cover rank, which is specified implicitly in the context as it is an ordering context. The following query reflects this semantics.

```
SELECT T.*
FROM books T
   INNER JOIN ctx_cover_STX C1
      ON(T.cover = C1.cover
      AND C1.user_name = CURRENT_USER)
WHERE T.instock
ORDER BY C1.rank;
```

Context 4: *The user does not prefer (wants to avoid) any science fiction books.*

This context may be defined as:

```
CREATE NEGATIVE CONTEXT ctxt_category_GQI (
   category integer,
   BINDING KEY (category)
      REFERENCES books(category)
) AS EQUIVALENCE WITH UNLISTED INCLUDED;

SET ACTIVE CONTEXT AS ctxt_category_GQI;
```

In response to Q_u, the actual query that is executed is given below. Rows in books, whose category exists as any of the contextual values of this context, are eliminated from the answer set.

```
SELECT T.*
FROM books T
WHERE T.category NOT IN (
          SELECT C1.category
          FROM ctxt_category_GQI C1)
          WHERE T.instock;
```

The basic contexts, which are not composed from other contexts, reflect in the actual executed query according to Table 2. This table shows whether an ORDER BY clause is necessary, and which type of join we need according to the context properties. We use the same symbols of the context classification as in Section 2 ([G] for negative context, [S] for positive context, etc.).

Next, we compose complex contexts from the above basic contexts. We start with the following context.

Context 5: *The user prefers books published in 2005, and then those published in 2006 before all other books. For the books that are similarly ranked, the user prefers hardcover books over books with paperback cover.*

This context may be viewed as the composition of ctxt_year_STI and ctxt_cover_STX. Therefore, we do not need to define a new

context. Conversely, we just need to set the active context appropriately to reflect to the desired context.

```
SET ACTIVE CONTEXT FOR user1
AS ctxt_year_STI, ctxt_cover_STX
WITH RANKING ORDER ctxt_year_STI, ctxt_cover_STX;
```

As a result of this combined context, queries to select tuples from books will work as if the query below was executed. First, the books in stock will be sorted based on the rank of the years, and then in case of ties, the cover type will be considered. This semantics is given by the following query rewrite.

```
SELECT T.*
FROM books T
   LEFT OUTER JOIN ctx_year_STI C1
      ON (T.year = C1.year
            AND C1.user_name = CURRENT_USER)
   INNER JOIN ctx_cover_STX C2
      ON (T.cover = C2.cover
            AND C2.user_name = CURRENT_USER)
WHERE T.stock
ORDER BY C1.rank, C2.rank;
```

Context Class	ORDER BY	Join Operation
GQN	X	NOT IN
SQN	X	LEFT OUTER JOIN
SQX	X	INNER JOIN
STN	√	LEFT OUTER JOIN
STX	√	INNER JOIN
SPN	√	LEFT OUTER JOIN
SPX	√	INNER JOIN

Table 2: The type of join used for each context class combination.

5. CHAMELEON PROOF-OF-CONCEPT INSTANTIATIONS

In this section, we illustrate how one can instantiate and realize specialized database servers using Chameleon. We begin with the first case study: privacy-aware databases. Then, we present spatial databases as our second case study. Finally, we conclude with two case studies that illustrate the ideas of context composition.

5.1 Realizing a Privacy Database Server

In this section, we show how we can limit disclosure, as what happens in Hippocratic Databases, using context awareness in Chameleon. In Table 3, we use the same patient table used in [33]. This table contains patient personal information.

Consider a healthcare facility that owns this data. Whenever a patient is admitted to the facility, he/she has to sign a privacy policy. The privacy policy specifies which information is to be released to which recipient. Moreover, the policy also specifies for which purposes the information is to be released. On an opt-in basis, the healthcare facility also allows patients to choose if they want any of their personal information to be released to other recipients. For instance, a nurse who is treating a patient is allowed to see the patient's name, age, and phone, but is not allowed to see his/her address for any reason. The patient may opt-in and choose that only his/her age is to be released to charity for solicitation.

pid	name	age	address	phone
1	Alice Adams	10	1 April Ave.	111-1111
2	Bob Blaney	20	2 Brooks Blvd.	222-2222
3	Carl Carson	30	3 Cricket Ct.	333-3333
4	David Daniels	40	4 Dogwood Dr.	444-4444

Table 3: The Patients Table.

Beside limited disclosure, limited retention is also modeled using context awareness. For simplicity, and without loss of generality, we assume that patient data is to be retained for 90 days only. By the end of this period, the patient data should have fulfilled the purposes for which the data has been collected. After this period, different recipients cannot retrieve the data.

It is important to make it clear that the patients in this context are the objects. Object contexts are the contexts of the patients. Moreover, users are those that use an application at the healthcare facility to retrieve patients' data. To model the above example of limiting the disclosure and retention of patients' data in Chameleon, we define the object contexts patient_privacy_pref and policy_signature as follows.

```
CREATE OBJECT CONTEXT patient_privacy_pref (
      recipient varchar(30),     purpose varchar(30),
      pid integer,               pid_pref boolean,
      name_pref boolean,         age_pref boolean,
      address_pref boolean,      phone_pref boolean,
   BINDING KEY(pid) REFERENCES patient(pid));

CREATE OBJECT CONTEXT policy_signature (
      pid integer,               sign_date date,
   BINDING KEY(pid) REFERENCES patient(pid));
```

Let the object context patient_privacy_pref contain the contextual data in Table 4. The following user context enforces the limited disclosure and limited retention of patients' data. Table 5 gives the context of three users. If the three users execute the query "SELECT * FROM patient;", they retrieve the data in Table 6.

```
CREATE POSITIVE CONTEXT identity_activity (
   job varchar(30),           activity varchar(30),
BINDING KEY(job, activity) REFERENCES
   patient_privacy_pref(recipient, purpose)
SUBSTITUTE patient(pid)
   WITH (CASE WHEN patient_privacy_pref.pid_pref
         AND today() <= policy_signature.sign_date + 90
         THEN patient.pid ELSE NULL)
SUBSTITUTE patient(name)
   WITH (CASE WHEN patient_privacy_pref.pid_pref
         AND today() <= policy_signature.sign_date + 90
         THEN patient.name ELSE NULL)
...
) AS EQUIVALENCE WITH UNLISTED EXCLUDED;
```

recipient	purpose	pid	pid_pref	name_pref	age_pref	address_pref	phone_pref
charity	solicitation	1	√	√	√	√	√
nurse	treatment	1	√	√	√	X	√
account clerk	billing	1	√	√	X	√	√
charity	solicitation	2	X	X	X	X	X
nurse	treatment	2	√	√	√	X	√
account clerk	billing	2	√	√	X	√	√
charity	solicitation	3	√	X	X	√	√
nurse	treatment	3	√	√	√	X	√
account clerk	billing	3	√	√	X	√	√
charity	solicitation	4	√	√	X	X	X
nurse	treatment	4	√	√	√	X	√
account clerk	billing	4	√	√	X	√	√

Table 4: The patient_privacy_pref object context.

user_name	job	activity
user1	charity	solicitation
user2	nurse	treatment
user3	account clerk	billing

Table 5: identity_activity contextual values

	pid	name	age	address	phone
u1	1	Alice Adams	10	1 April Ave.	111-1111
	3			3 Cricket Ct.	333-3333
	4	David Daniels			
u2	1	Alice Adams	10		111-1111
	2	Bob Blaney	20		222-2222
	3	Carl Carson	30		333-3333
	4	David Daniels	40		444-4444
u3	1	Alice Adams		1 April Ave.	111-1111
	2	Bob Blaney		2 Brooks Blvd.	222-2222
	3	Carl Carson		3 Cricket Ct.	333-3333
	4	David Daniels		4 Dogwood Dr.	444-4444

Table 6: Result of "SELECT * FROM patient;" for all users u1, u2, and u3.

5.2 Realizing a Spatial Database Server

Spatial databases are optimized to store and query data related to objects in space. This type of databases has more complex geometrical data types, e.g., points, lines, and rectangles.

Consider a real-estate database containing information about houses. The houses table has the following schema: houses (id, bedrooms, price, city). An application developer is interested in providing some spatial queries to this database, but has no privileges to add the location of the house to this table. An object context is created to add the location of houses.

5.2.1 Range Queries

Let the user context be the willingness to buy a house in certain regions. Hence, a user context is created in Chameleon to declare that only houses contained in relevant regions are to be returned.

The definitions of the object and user contexts, house_loc and houses_in_region, respectively, are given below. The function "contained" retrieves any house with location (x, y) that exist with the rectangular region (x1, y1, x2, y2). The binding between object and user contexts is through the scalar function "contained" that retrieves only the database objects within the query issuer's range context. Notice that there is no prioritization for the objects within the range, and hence the EQUIVALENCE keyword specifies the lack of any ordering.

```
CREATE OBJECT CONTEXT house_loc (
        id integer,
        x integer,          y integer,
  PRIMARY KEY(id),
  BINDING KEY id REFERENCES  houses(id));

CREATE POSITIVE CONTEXT houses_in_region (
        x1 integer,         y1 integer,
        x2 integer,         y2 integer,
  BINDING KEY() REFERENCES house_loc
  WITH contained (house_loc.x, house_loc.y, x1, y1, x2, y2)
) AS EQUIVALENCE WITH UNLISTED EXCLUDED;
```

5.2.2 Nearest Neighbor Queries

Another class of queries in spatial databases is the nearest neighbor query. In this class, the user wants to retrieve the object that is nearest to a pivot location. An extension to this class of queries is the k nearest-neighbors query. The answer of this query is the k objects that are nearest to the pivot location. In the real estate database, a user willing to retrieve the houses listed by proximity to a point may declare her context as follows:

```
CREATE POSITIVE CONTEXT nearby_houses (
        x integer,          y integer,
  BINDING KEY() REFERENCES house_loc
  WITH true
 ) AS TOTAL ORDER USING
        dist(x, y, house_loc.x, house_loc.y)
WITH UNLISTED EXCLUDED;
```

Notice that the clause WITH UNLISTED EXCLUDED can be omitted since all the houses are totally ordered based on distance.

The equivalent SQL query with the awareness of this context would be:

```
SELECT T.*
FROM houses T
  INNER JOIN house_loc OC1
        ON(T.id = OC1.id),
  nearby_houses C2
ORDER BY dist(C2.x, C2.y, OC1.x, OC1.y)
```

Notice that for finding nearest-neighbors to a user's location, the binding between the database objects and the user's focal point is via a total order based on a scalar function "distance".

5.3 Combining Contexts

5.3.1 Skylines

Skyline queries emerge in spatial databases. Assume that user2 wants to buy a house that is close to his work in downtown and that is also cheap (or at least reasonable) in price. Since it is not easy to combine such preferences in a ranking expression, user2 decides to select from the skyline houses.

Such context is defined as the composition of several contexts, namely houses_in_region, nearby_houses, and the context price already in the houses table. The first context will include a bounding box representing downtown. The second and third contexts will be used to compute the skyline. This composition is instantiated by setting the active context as follows:

```
SET ACTIVE CONTEXT FOR user2
   AS houses_in_region, nearby_houses
WITH SKYLINE OF nearby_houses.rank MIN,
                houses.price MIN;
```

Notice that after defining the houses in certain areas, and then defining the closeness to pivot points, the SET ACTIVE CONTEXTS combines both of these contexts (the range context and the nearest-neighbors context) together along with an object context (price) that is part of the house relation to get the SKYLINE of distance and price. This illustrates a more complex usage of contexts to answer conjunctions of spatial predicates.

5.3.2 Location-aware Privacy

Consider an application scenario where a query issuer, e.g., a doctor, may be allowed to access a database object's record, e.g., patient's record, only when the doctor is in the hospital premises. Otherwise, the doctor is not allowed to access the records.

In order to realize a location-aware privacy database server, we make use of the two contexts patient_privacy_pref and identity_activity that we define in Section 5.1 to realize the privacy context. For the location context, we make use of the two simple object and user contexts (valid_location and current_location, respectively). In this example, the location is modeled by a string value that gives a high-level description of the user's or the object's location (in contrast to physical coordinate locations). The mapping from the physical location to the named location is skipped here for simplicity.

```
CREATE OBJECT CONTEXT valid_location (
   pid integer,
   location varchar(30),
   BINDING KEY (pid) REFERENCES patient(pid) );

CREATE POSITIVE CONTEXT current_location (
    location varchar(30),
    BINDING KEY (location)
        REFERENCES valid_location (location)
) AS EQUIVALENCE WITH UNLISTED EXCLUDED;

SET ACTIVE CONTEXT identity_activity, current_location;
```

Notice that only the user contexts are listed since the binding activates the corresponding object contexts.

5.3.3 Location-aware and Time-aware Privacy

The example below gives a more complex context composition of the identity, location, and time contexts for both the database objects and the query issuers to realize a database server that provides both location-aware and time-aware privacy. This server would be useful in guaranteeing that, for example, a doctor may be allowed to access a patient's record only when the doctor is in the hospital but not after the hospital's regular hours.

We make use of the contexts patient_privacy_pref and identity_activity (defined in Section 5.1) to realize the privacy context, the contexts valid_location and current_location (defined in Section 5.3.2) to realize the location context, and the temporal contexts valid_time, current_time_not_expired, and current_valid_time, defined below.

```
CREATE OBJECT CONTEXT policy_signature (
   pid integer,
   sign_date date,        expire_date date,
   BINDING KEY(pid) REFERENCES patient(pid) );

CREATE OBJECT CONTEXT valid_time (
   pid integer,
   from_time date,        to_time date,
   BINDING KEY(pid) REFERENCES patient(pid) );

CREATE POSITIVE CONTEXT current_time_not_expired (
  BINDING KEY() REFERENCES patient
     WITH today() >= policy_signature.sign_date
     AND today() <= policy_signature.expire_date)

CREATE POSITIVE CONTEXT current_time_valid (
  BINDING KEY() REFERENCES patient
     WITH now() >= valid_time.from_time
     AND now() <= valid_time.to_time)

SET ACTIVE CONTEXT identity_activity, current_location,
current_time_not_expired, current_time_valid;
```

Notice that the user context current_time_not_expired provides limited retention, i.e., that data is made available only for the duration agreed upon by the data owner.

6. RELATED WORK

There have been several definitions of context and context-awareness (e.g., see [4, 6, 7, 17, 25, 40, 43, 44]). Most of these definitions define the context in terms of examples with special emphasis on the location context. Similarly, there have been several definitions of context-aware applications that include various synonyms, e.g., adaptive applications [44], reactive applications [16], responsive applications [19], situated applications [25], contented-sensitive applications [42], and environment directed applications [21]. In this paper, we adhere with the most formal definitions given in [17]. Recently, there has been interest in adding the context-awareness to relational database systems and query processors (e.g., see [30, 46]). However, the main focus is either on the modeling of the context

information and how to integrate it into the query definition, or on very specific examples that consider only one type of context. None of the previous work have discussed or proposed a full-fledge realization of context-awareness inside a DBMS.

There has been several work for presenting preferences in terms of relational calculus, first order logic, and query languages (e.g., see [14, 28, 32, 50]). In terms of query processing, there are two extremes for preference-aware queries, namely, top-k and skyline queries. Top-k queries have been well studied in various fields (e.g., [9, 12, 20, 38]). Also, there have been numerous algorithms for embedding top-k queries into database operators (e.g., see [8, 13, 22, 26, 34]). On the other hand, the term skyline queries has been coined in the database literature [5] to refer to the secondary storage version of the maximal vector set problem [31, 36]. Due to its practicality, various versions of skyline queries have been studied in the literature, e.g., sorted data [15], partially-ordered domains [10], high-dimensional data (e.g., [11, 41, 49, 52, 53]), progressive and online computations (e.g., [29, 39, 47]), sliding window [35, 48], continuous skyline computations [24, 37, 51], mobile ad-hoc networks [23], spatial skylines [45], and data mining [27]. Unlike the case for top-k queries, there is no previous work in integrating skyline queries at the core of query operators or database systems.

7. RESEARCH CHALLENGES AND CONCLUDING REMARKS

A working demonstration of the Chameleon context-aware database management system is currently available based on extensions to PostgreSQL. In the resulting context-aware DBMS, Chameleon, we also implement several operators to combine multiple contexts, mainly, the SkylineJoin, the RankJoin, and the FilterMark operators. Figure 8 illustrates the components we modified in PostgreSQL to realize Chameleon.

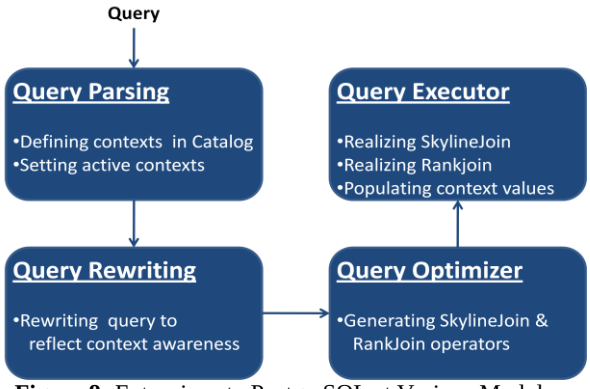


Figure 8: Extensions to PostgreSQL at Various Modules.

Based on this prototype of Chameleon, the authors have identified the following research challenges:

Performance: In the authors' opinion, the introduction of context-aware database management systems as outlined in this paper (and as depicted in Figure 1) is a profound and an important step. The declarative approach in defining the queries in relational databases (in contrast to the procedural approach in network databases) was one of the main factors that made the relational model prevail. Analogously, the declarative approach in defining contexts in context-aware database management systems can have a strong impact. Over thirty five years of efficient implementation and tuning has made the relational model overcome the efficiency hurdle. Similarly, efficient realization is the main challenge in context-aware database management systems. Efficient realization and execution are the authors' main focus for future work.

Dynamic Contexts: Another important challenge is that of dynamic contexts. So far, what Chameleon offers is static contexts. In many application scenarios, changes take place in the contexts, e.g., some active contexts may become inactive, inactive ones may become active, or new contexts get introduced. Another form of change is that the contextual values themselves within a context may change, e.g., the surrounding temperature may change or the location of a moving object may change, etc. These changes may affect the query being executed. This is similar in spirit to mid-query reoptimization [54]. However, the difference is that when the contexts change, the system may need to augment the query being executed by additional predicates that reflect that change in contexts.

Expressiveness and Completeness: Finally, issues related to the expressiveness and completeness of the context-aware model presented in this paper need to be studied.

8. REFERENCES

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *VLDB*, Hong Kong, China, August 2002.

[2] J. Padma, Y.N. Silva, M.U. Arshad, W.G. Aref: Hippocratic PostgreSQL. In ICDE, Shanghai, (Mar. 2009) 1555-1558.

[3] H.G. Elmongui, W.G. Aref, and M. Mokbel. Chameleon: Context-Awareness inside DBMS. In ICDE, Shanghai, (Mar. 2009) 1335-1338.

[4] Merriam-Webster Dictionary. http://www.m-w.com/.

[5] S. Börzsönyi, D. Kossmann, and K. Stocker. The Skyline Operator. In ICDE, 2001.

[6] P. Brown. The Stick-e document: a framework for creating context-aware applications. Electronic Publishing, 8(2&3), 1996.

[7] P. J. Brown, J. D. Bovey, and X. Chen. Context-aware Applications: From the laboratory to the marketplace. IEEE Personal Communications, 4(5), 1997.

[8] N. Bruno, S. Chaudhuri, and L. Gravano. Top-k Selection Queries over Relational Databases: Mapping Strategies and Performance Evaluation. TODS, 27(2), 2002.

[9] N. Bruno, L. Gravano, and A. Marian. Evaluating Top-k Queries over Web-Accessible Databases. In ICDE, 2002.

[10] C.Y. Chan, P.-K. Eng, and K.-L. Tan. Stratified Computation of Skylines with Partially-Ordered Domains. In SIGMOD, 2005.

[11] C. Y. Chan, H. V. Jagadish, K.-L. Tan, A. K. H. Tung, and Z. Zhang. Finding k-Dominant Skylines in High Dimensional Space. In SIGMOD, 2006.

[12] K. C.-C. Chang and S. won Hwang. Minimal Probing: Supporting Expensive Predicates for Top-k Queries. In SIGMOD, 2002.

[13] Y.-C. Chang, L. D. Bergman, V. Castelli, C.-S. Li, M.-L. Lo, and J. R. Smith. The Onion Technique: Indexing for Linear Optimization Queries. In SIGMOD, 2000.

[14] J. Chomicki. Preference Formulas in Relational Queries. TODS, 28(4), 2003.

[15] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with Presorting. In ICDE, 2003.

[16] J. R. Cooperstock, K. Tanikoshi, G. Beirne, T. Narine, and W. Buxton. Evolution of a Reactive Environment. In Proceeding of the International Conference on Human Factors in Computing Systems, CHI, 1995.

[17] A.K. Dey and G.D. Abowd. Towards a better understanding of context and context-awareness. In Workshop on the What, Who, Where, When, and How of Context-Awareness, CHI, 2000.

[18] H.G. Elmongui and W.G. Aref. Skyline-Aware Join Operator. Tech. Rep. CSD TR 08-007, Purdue Univ., 2008.

[19] S. Elrod, G. Hall, R. Costanza, M. Dixon, and J. des Rivières. Responsive Office Environments. Communications of ACM, 36(7), 1993.

[20] R. Fagin, R. Kumar, and D. Sivakumar. Comparing Top k Lists. SIAM Journal on Discrete Mathematics, 17(1), 2003.

[21] S. Fickas, G. Kortuem, and Z. Segall. Software Organization for Dynamic and Adaptable Wearable Systems. In International Symposium on Wearable Computers, 1997.

[22] V. Hristidis, N. Koudas, and Y. Papakonstantinou. PREFER: A System for the Efficient Execution of Multi-parametric Ranked Queries. In SIGMOD, 2001.

[23] Z. Huang, C. S. Jensen, H. Lu, and B. C. Ooi. Skyline Queries Against Mobile Lightweight Devices in MANETs. In ICDE, 2006.

[24] Z. Huang, H. Lu, B.C. Ooi, and A. K. Tung. Continuous Skyline Queries for Moving Objects. TKDE, 18(12), 2006.

[25] R. Hull, P. Neaves, and J. Bedford-Roberts. Towards Situated Computing. In International Symposium on Wearable Computers, 1997.

[26] I.F. Ilyas, W.G. Aref, A.K. Elmagarmid, H.G. Elmongui, R. Shah, and J.S. Vitter. Adaptive Rank-Aware Query Optimization in Relational Databases. TODS, 31(4), 2006.

[27] W. Jin, J. Han, and M. Ester. Mining Thick Skylines over Large Databases. In PKDD, 2004.

[28] W. Kießling. Foundations of Preferences in Database Systems. In VLDB, 2002.

[29] D. Kossmann, F. Ramsak, and S. Rost. Shooting Stars in the Sky: An Online Algorithm for Skyline Queries. In VLDB, 2002.

[30] G. Koutrika and Y. E. Ioannidis. Personalized Queries under a Generalized Preference Model. In ICDE, 2005.

[31] H. T. Kung, F. Luccio, and F. P. Preparata. On Finding the Maxima of a Set of Vectors. Journal of ACM, 22(4), 1975.

[32] M. Lacroix and P. Lavency. Preferences: Putting More Knowledge into Queries. In VLDB, 1987.

[33] K. LeFevre, R. Agrawal, V. Ercegovac, R. Ramakrishnan, Y. Xuy, and D. DeWitt. Limiting disclosure in Hippocratic databases. In VLDB, 2004.

[34] C. Li, K. C.-C. Chang, I. F. Ilyas, and S. Song. RankSQL: Query Algebra and Optimization for Relational Top-k Queries. In SIGMOD, 2005.

[35] X. Lin, Y. Yuan, W. Wang, and H. Lu. Stabbing the Sky: Efficient Skyline Computation over Sliding Windows. In ICDE, 2005.

[36] J. Matousek. Computing Dominances in E^n. Information Processing Letters, 38(5), 1991.

[37] M. D. Morse, J. M. Patel, and W. I. Grosky. Efficient Continuous Skyline Computation. In ICDE, 2006.

[38] A. Natsev, Y.-C. Chang, J. R. Smith, C.-S. Li, and J. S. Vitter. Supporting Incremental Join Queries on Ranked Inputs. In VLDB, 2001.

[39] D. Papadias, Y. Tao, G. Fu, and B. Seeger. Progressive skyline computation in database systems. TODS, 30(1), 2005.

[40] J. Pascoe. Adding Generic Contextual Capabilities to Wearable Computers. In International Symposium on Wearable Computers, 1998.

[41] J. Pei, W. Jin, M. Ester, and Y. Tao. Catching the Best Views of Skyline: A Semantic Approach Based on Decisive Subspaces. In VLDB, 2005.

[42] J. Rekimoto, Y. Ayatsuka, and K. Hayashi. Augment-able Reality: Situated Communication Through Physical and Digital Spaces. In Intl. Symp. on Wearable Computers, 1998.

[43] T. Rodden, K. Chervest, N. Davies, and A. Dix. Exploiting Context in HCI design for Mobile Systems. In HCI, 1998.

[44] B.N. Schilit and M.M. Theimer. Disseminating Active Map Information to Mobile Hosts. IEEE Network, 8(5), 1994.

[45] M. Sharifzadeh and C. Shahabi. The Spatial Skyline Queries. In VLDB, 2006.

[46] K. Stefanidis, E. Pitoura, and P. Vassiliadis. Adding Context to Preferences. In ICDE, 2007.

[47] K.-L. Tan, P.-K. Eng, and B. C. Ooi. Efficient Progressive Skyline Computation. In VLDB, 2001.

[48] Y. Tao and D. Papadias. Maintaining Sliding Window Skylines on Data Streams. TKDE, 18(2), 2006.

[49] Y. Tao, X. Xiao, and J. Pei. SUBSKY: Efficient Computation of Skylines in Subspaces. In ICDE, 2006.

[50] G.K. Werner Kießling. Preference SQL - Design, Implementation, Experiences. In VLDB, 2002.

[51] T. Xia and D. Zhang. Refreshing the Sky: The Compressed Skycube with Efficient Support for Frequent Updates. In SIGMOD, 2006.

[52] Y. Yuan, X. Lin, Q. Liu, W. Wang, J. X. Yu, and Q. Zhang. Efficient Computation of the Skyline Cube. In VLDB, 2005.

[53] Z. Zhang, X. Guo, H. Lu, A. K. H. Tung, and N. Wang. Discovering Strong Skyline Points in High Dimensional Spaces. In CIKM, 2005.

[54] N. Kabra, D.J. DeWitt: Efficient Mid-Query Re-Optimization of Sub-Optimal Query Execution Plans. SIGMOD Conference 1998: 106-117.

Privacy-Enabling Abstraction and Obfuscation Techniques for 3D City Models (Position Paper)

Martin Kada, Michael Peter, Dieter Fritsch
Institute for Photogrammetry, University of Stuttgart
Geschwister-Scholl-Straße 24D
70174 Stuttgart, Germany
++49 (0)711/685-83386
[firstname.lastname]@ifp.uni-stuttgart.de

Oliver Siemoneit, Christoph Hubig
Institute of Philosophy, University of Stuttgart
Seidenstraße 36
70174 Stuttgart, Germany
++49 (0)711/685-82491
[firstname.lastname]@philo.uni-stuttgart.de

ABSTRACT
Privacy and security issues within geospatial information systems are of growing public and scientific interest. Especially with the launch of Google Street View and Google Earth, geospatial data has come to the attention of the public, thereby not only raising support for these technologies, but also massive concerns. It is the duty of science to pick up today's uprising debates and to help structuring them, providing clarifications and different solutions. Thus, the aim of this paper is to contribute in form of an interdisciplinary discussion about privacy issues, both from a philosophical and an engineering point of view. Privacy and its importance are outlined as well as different privacy issues raised concerning the nowadays so popular 3D city models. In addition, technical solutions are shown which allow data providers to preserve privacy, but that won't interfere with the advancements of these technologies.

Categories and Subject Descriptors
K4.1 [Computers and Society]: Public Policy Issues - Privacy

General Terms
Human factors, legal aspects, algorithms

Keywords
Ethics, privacy, privacy-enhancing technologies, 3D city models, street views

1. INTRODUCTION
Technological developments usually come forth with a vast amount of new opportunities. But at the same time, technological innovations are also prone to novel, unknown problems and threats–for its latter users and/or society in general. It is the unavoidable, janus-faced nature of technology, which has also recently drawn a lot of attention to geospatial information systems and services. Especially in a lot of European countries–where privacy and data protection laws are far stricter than in the United States–some early kind of geospatial information system has proven to be highly controversial. The launch of Google Street View has caused many citizens to issue complaints to government officials about the project thereby claiming that it is a massive intrusion upon privacy and thus a violation of existing data privacy laws [1]. In this quite emotional, heated and sometimes even irrational debate, it is the job of the sciences to pick up the raised questions, to think about them, to analyze, and to restructure them in joint, interdisciplinary research and finally to think of adequate solutions. It is the main aim of this paper, to make a basic contribution to this debate and to shed some light on fundamental questions. Therefore, in section 2, it is first elaborated on what privacy is at all and which role privacy plays in western societies. In section 3, different privacy issues within geospatial information systems are identified and outlined. Section 4 presents techniques that offer the potential to better preserve privacy in 3D city models, but also discusses their limitations. Section 5 concludes our work, which we regard as a stepping stone to future research directions.

2. PRIVACY AND ITS IMPORTANCE
Privacy and the right of privacy is central for all liberal, egalitarian, and democratic societies [2][3]. Privacy assures personal freedom and autonomy, guarantees freedom from governmental interventions, or other societal institutions, parties or persons and thus allows a person to build his own individual scheme of life. It is privacy that establishes a sphere of non-intervention which is also crucial for self-fulfillment and the development of a personal identity. So-called *decisional privacy* concerns therefore basic decisions of a person about who he wants to be and how he wants to live. Decisional privacy is the core of one's personal freedom and the possibility to form one's own authentic identity. It is also the core of political freedom in the form of the absence of interferences with the sovereignty (negative freedom as "freedom from") and the assistance in fulfilling one's own potential (positive freedom as "freedom to").

On the contrary, so-called *informational privacy* deals with the fact that a person wants to be in control of personal information about intimacies of his life. In this clearly information-based or

knowledge-based conception of privacy, privacy intrusions are defined therefore as situations in which personal information is collected or disseminated without consent of the person who is topic of the information. Informational privacy is crucial for regulating personal relationships and establishing different social roles one plays in society: "If everyone knew everything about everyone else, differentiated relations and self-presentation would no longer be possible, nor would autonomy and the freedom to determine one's own life" [3].

However purely information-based conceptions of privacy are clearly flawed: There are also other privacy violations, which are not of a cognitive nature but of a physical one: *Local privacy* is the right of a person to restrict physical access of others to his body, his personal belongings and his home. Local privacy assures therefore a sphere of non-intervention, a protected, secured, private place or shelter. This definition of privacy corresponds well with the famous description of Samuel D. Warren and Louis D. Brandeis of the right of privacy as the right to be left alone [12]. Not only is it important to be left alone from the gaze and opinions of others, but also the right to control physical interference by others into one's private affairs.

To conclude, we could say, with Ferdinand Schoeman, that "a person has privacy to the extent that others have limited access to information about him, limited access to the intimacies of his life, or limited access to his thoughts or his body" [4]. The right of privacy is then the right of a person to be protected against intrusions (negative form of privacy as being free from) and to be able to control cognitive or physical access to his personal things and affairs (positive form of privacy as being able to decide freely to). Thus, privacy allows for inner and outer freedom of an individual, helps building and assuring the personal integrity and autonomy, helps protecting his reputation, is enabling different forms of social self-representation in different social contexts.

If you have a look at the history of privacy, it becomes obvious, that what counts as "public" or "private" depends largely on the social tradition and varies from culture to culture [3]. Privacy is therefore of "conventional nature" only and subject to an on-going societal negotiation process. The personal right of privacy is delimited and overridden by other rights and competing moral principals so as to protect interests and rights of other parties or of society in general. (E.g. at the workplace, privacy is neither totally free from restrictions nor does a contract of employment nullify privacy claims at all. Or as Anders J. Persson and Sven Ove Hansson put it, taking another form of contractual relation as example: Having a rental contract will give the owner of the house the right to enter the house for certain purposes, but not to open closets and read private papers that are kept in there [5].)

All in all, from a theoretical point of view, this societal balancing and negotiation process could be best described by the concept of so-called reflective equilibrium [6]. On the one hand, certain "given" values or norms do restrict our social practices. On the other hand, looking at the practical effects of these norms, we do also change certain norms and values we consider as too restrict, inadequate or outdated. In our case this means: On the one hand personal privacy rights are weighed against other rights restricting/enlarging personal privacy. On the other hand existing rights and social norms are also changed, because we are not willing to accept them anymore (since they restrict privacy too much).

3. PRIVACY ISSUES IN GEOSPATIAL INFORMATION SYSTEMS

According to the directive 2007/2/EC of the European Parliament and Council, geospatial data is "any data with a direct or indirect reference to a specific location or geographical area" thereby often describing a spatial object which is further defined as an "abstract representation of a real-world phenomenon related to a specific location or geographical area" [7]. Geospatial data is therefore only object-related data and not initially subject to data protection laws [9]. However–under certain circumstances–geospatial data could become personal data [8][9]. This is the case, if 1) photos or photo-realistic views/models of spatial objects (i.e. a building or an estate) could be easily located by geo-coordinates and thus easily matched to its owner or residents and/or if 2) the data is–to put it more generally–able to describe personal or factual affairs [8][9]. In these cases, geospatial data is also subject to data protection laws. Then, the collection, storage, processing and dissemination of the data is only allowed, if the interests of the individuals, which are subject of the data, are not harmed and/or are not superseded by other rights and interests (such as homeland security) [8].

Picking up the above mentioned "dimensions" of privacy, privacy intrusions in the realm of geospatial data are of cognitive nature only and therefore mainly intrusions on informational privacy (with possible effects on decisional and local privacy in the future):

1) Geospatial data showing faces of people, license plates of cars (as Google Street View does) could be seen as problematic, since it conveys a lot of information on personal affairs, such as personal habits, preferences, circumstances [8][10]. Even obfuscating faces or license plates is often not sufficient, as a lot of things still remain recognizable because of other distinctive, individual characteristics [10].

2) The same is true for showing house numbers and detailed, photo-realistic images or representation of spatial objects, since it tells a lot about personal circumstances and thus could allow for geo-marketing or scoring of creditworthiness [8][10].

3) Especially Google Street View is criticized for having a "privileged view" on the spatial object: Pictures are taken at the height of 2.5m and not at the height of the eyes of a pedestrian, thus allow to look inside an estate or home–a per se secured, protected, intimate space [10].

However, data protection officials also agree on this: If the spatial object is obfuscated or the presentation of the spatial object is of abstract manner only, no interests of individuals are violated [8]. Therefore, in the following, different methods and techniques are to be presented and outlined, which meet these requirements and "remove" certain privacy issues. However, one has to bear in mind, that the core of data protection is not met by that: Especially in Germany, the data protection officials want to force Google not just to obfuscate images and grant individuals the right to get certain pictures removed from the database, but also to delete all non-obfuscated raw-data (so as not to be able to use the data anymore for e.g. commercial purposes in countries where data protection laws are not as strict as in Europe) [11]. And indeed this is the case: Google has collected data in Germany but has transferred all data for storing and further processing to the

US thereby not willing to delete the raw material and thus the data still being open for abuse and possible privacy violations [11].

All in all, it should be highly appreciated, that privacy and geospatial data is discussed more and more on a broad public basis. It is the duty of society in general to decide, which technology to adopt (respectively how to adopt a technology): We need not do all the things we are (technically) capable to do. It is the job of philosophy and the engineering sciences to accompany, support and guide these debates, to clarify things and to outline and provide different solutions.

4. PRIVACY-ENABLING ABSTRACTION AND OBFUSCATION TECHNIQUES

As already mentioned, the obfuscation of faces and license plates still leaves a lot of information in the image, so that a person or one person's property could still be recognizable. The distinct characteristics could be very small or unusual, which makes it nearly impossible to automatically detect and remove them all by processing a single image at a time. Even if it would be possible, the resulting images would depict scenes where large parts are blurred or missing. As such images are not attractive to anyone this is obviously not a viable solution. The only alternative is therefore to use image sequences that show the same scene at different times and/or from different angles. Then the critical objects are hopefully gone or at least are located in a different part of the image and have cleared the view to the formerly occluded area. Multiple images allow for an image fusion to produce new ones without people and private objects.

Most comparable work has been done for the automatic generation of façade textures from terrestrial images, where occlusions from cars and pedestrians are avoided by a filtering of multiple images. Böhm [13] e.g. blends per-pixel registered images in a color clustering approach in order to synthesize occlusion-free texture images for building façades (see Figure 1 left). By only capturing a handful of images from multiple stations or a sequence of images from one point, both moving and static objects that are in front of the façade can be completely eliminated. However, each pixel must exactly point to the same planar part of the façade as the corresponding pixels of the other images. Such image correspondences can be reliably determined by the SIFT operator [14], which has also been implemented to run in real-time [23]. Although an automatic retouching of façade images is only possible if the underlying façade geometry is known, the necessary methods for a reconstruction from stereo imagery and laser scanning data at street level has long been shown (see e.g. [20][21][22]). And as recent reports have stated, the Google Street View vehicles of Google have been spotted with laser scanners mounted on the roof.

Until now, we have only regarded objects that are in front of the façade and not on the façade. This applies to house numbers, name plates and billboards. And although stores, firms and companies place them by the majority for advertising purposes, private persons and small firms might feel their privacy violated by this unwanted publicity. Such objects could be detectable by optical character recognition (OCR), which has reached a level where letters and numbers are reliably recognized. The question remains what to do with these areas? In contrast to persons and cars, a blurring of the characters would in most cases be sufficient to make them unrecognizable. Again, such an approach is not very appealing as it degrades the quality of the façade textures. Better would be to retouch these areas by copying similar parts of the façade image (see Figure 1 right, bottom).

Once the objects in front of the building have been eliminated and the façades been cleared, the next level of anonymization is to remove what can be seen of the interior of the building. The major intrusion into private homes can be expected coming from the windows. To counteract this, the glass parts could be grayed out and given a bright streak of reflected light to keep a realistic appearance. Another option would be to store the semantic information, so that a visualization application can adapt the window glass to better reflect the environment and weather conditions. However, before the relevant pixels can be altered, the

Figure 1: Left: Occlusion-free texture (bottom) by multiple image fusion. Right: Two abstraction levels (1st image, 2nd geometry) from a photo-realistic 3D building model (top).

locations and shapes of the windows must be detected. Several publications have addressed this problem. By using the Förstner operator [15], Mayer matches façade images to a database containing images of common window types. This enables the identification of the position and dimensions of the windows [16]. Ripperda and Brenner reconstruct the arrangement of doors and windows in a stochastic Reversible jump Markov Chain Monte Carlo process using formal grammars of façades [17]. Becker and Haala detect 3D edges in image pairs to do a hypothesis test on the existence of glazing bars and fanlights of windows and doors [18]. Also Wenzel et al. detects repetitive structures in facade images by using the SIFT operator in conjunction with a heuristic search method [19].

A photorealistic visualization might not always be necessary in all applications. Döllner and Kyprianidis e.g. present an automatic image abstraction approach that, applied to image sequences of 3D city models, results in a realistic, but cartoon-like presentation of virtual environments [24] (see Figure 1 right, center). On the one hand, such a presentation of real-life objects features enough details to recognize the spatial situation, but on the other hand changes enough to make re-identification of persons impossible and the judgment on people's living conditions inconclusive. In a last image abstraction step, the facades and roofs could be colored in a single color only.

The abstraction of (façade) images is only one aspect concerning the privacy of geospatial data. Another is the geometry, which can be regarded both by single buildings, but also by their arrangement into building blocks.

Over the last decade, quite some work has been dedicated to the simplification of 3D building models for cartographic purposes (e.g. [25][26][27][28]). In contrast to surface simplification algorithms known from the field of computer graphics, these algorithms are specifically designed for buildings. They strictly maintain global symmetries and enforce geometric properties like the co-planarity, parallelism and rectangularity of façade walls with the purpose to avoid that they become tilted in the simplification process.

Because not every application needs highly detailed models, as long as the final outcome is the same, we suggest using such a geometric abstraction as a way to protect peoples' privacy. During route guidance, e.g., only those details that assure a high recognition rate of the landmarks and that are necessary for a particular route should be presented to the user. Objects that are of no interest in the application's context must not be exposed to the public in every detail (see Figure 1 right, bottom), because a highly detailed representation could reveal private information about someone's living condition. For the driver of a vehicle, the single colored flat façade of a building has the same informative value as a detailed façade with windows, dormers, doors, etc. However, if highly detailed façade models are a requirement, their appearance could be obfuscated by a generalization that changes the number and arrangement of façade elements [26].

The above mentioned generalization algorithms try to maintain the object's shape characteristics as best as possible. This is especially useful for landmarks and other buildings with unusual architectures. For residential buildings, which are generally of higher concern regarding privacy issues, an even stronger shape simplification can be achieved by the use of 3D building symbols or standard roof shapes (see e.g. [29][30]).

At this point, we want to leave single buildings behind us and take a look at spatial situations with several buildings. There are two alternatives for an abstraction: typification and aggregation. Typification is a generalization technique where the spatial situation is analyzed to detect similar objects and their arrangement. Then the number of objects is reduced while maintaining the global appearance. For example, two buildings in a row of five similar looking houses could be removed and the remaining three re-located and increased in size to fill the idle space. Traditionally, such a technique is used to make room in a map when its scale changes and all objects won't fit anymore in the same space. This technique, however, could well be used to obfuscate a spatial situation or even hide buildings that are at risk concerning their security. As typification of 3D city models is really only a 2D problem, an algorithm like the one described in [31] could be used.

While such an abstraction approach still results in models that comprise of several entities, the aggregation operation replaces all buildings with a building block. Anders shows e.g. an approach that works on 3D building models [32]. Glander and Döllner aggregate building blocks while highlighting landmarks [33]. Such approaches could be context-sensitive, thus presenting only the detailed information that is vital to the task. The remaining objects are simplified to protect the privacy of residents, owners, public and private facilities.

5. CONCLUSION

In this paper, we gave an in-depth discussion on privacy in general and privacy issues in special within geospatial information systems. Furthermore, different abstraction and obfuscation techniques have been presented helping to circumvent possible (re-) identification of people and their living conditions and shield institutions that are at risk from prying eyes, which otherwise would be possible from highly accurate and detailed 3D city models.

6. REFERENCES

[1] Spiegel Online International. 2008. Privacy Concerns: German Towns Saying 'Nein' to Google Street View. URL=http://www.spiegel.de/international/germany/0,1518,581177,00.html.

[2] Rössler, B. 2004. The Value of Privacy. Wiley.

[3] Rössler, B. 2006. New Ways of Thinking about Privacy. In Phillips, A., Honig, B. and Dryzek, J. (eds.) The Oxford Handbook of Political Theory. Oxford University Press. pp. 694-712.

[4] Schoeman, F. 1984. Philosophical Dimensions of Privacy. Cambridge University Press.

[5] Persson, A.J., and Hansson, S.O. 2003. Privacy at Work: Ethical Criteria. In Journal of Business Ethics, Vol. 42., pp. 59-70.

[6] Norman, D. 2008. Reflective Equilibrium. In Zalta, E. N. (ed.) The Stanford Encyclopedia of Philosophy (Fall 2008 Edition). URL=http://plato.stanford.edu/archives/fall2008/entries/reflective-equilibrium/.

[7] Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE).

2007. URL=http://www.emwis.net/documents/fol962872/euro_legislation/stl078838.

[8] Resolution of the German top data protection authority "Düsseldorfer Kreis" on the provision of digital street views especially in the internet (German). 2008. URL=http://www.bfdi.bund.de/cln_118/SharedDocs/Publikationen/Entschliessungssammlung/DuesseldorferKreis/141108DigitaleStrassenansichten.html?nn=409242.

[9] Forgó, N., Krügel, T., and Reiners, N. 2008. Expert's report on geospatial data and data protection (German). URL=http://www.iri.uni-hannover.de/tl_files/pdf/Gutachten%20GEODAT.pdf.

[10] Privacy International. 2009. PI files complaint about Google Street View. URL=http://www.privacyinternational.org/article.shtml?cmd%5B347%5D=x-347-564039.

[11] Spiegel Online International. 2009. Protecting Privacy: Hamburg reaches Deal with Google on Street View. URL=http://www.spiegel.de/international/zeitgeist/0,1518,626075,00.html.

[12] Warren, S.D., and Brandeis, L.D. 1890. The Right of Privacy. In Harvard Law Review, Vol. 4, No. 5, pp. 193-220.

[13] Böhm, J. 2004. Multi-Image Fusion for Occlusion-Free Façade Texturing. In The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXV, Part B, Istanbul, Turkey.

[14] Lowe, D.G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. In International Journal of Computer Vision, 60-2, 91-110.

[15] Förstner, W., and Gülch, E. 1987. A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features. In ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, 281-305.

[16] Reznik, S., and Mayer, H. 2007. Implicit Shape Models, Model Selection and Plane Sweeping for 3D Façade Interpretation. In the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVI, 3/W49A, 173-178.

[17] Ripperda, N., and Brenner, C. 2009. Application of a Formal Grammar to Façade Reconstruction in Semiautomatic and Automatic Environments. In Proceedings of the 12th AGILE Conference on GIScience, Hanover, Germany.

[18] Becker, S., and Haala, N. 2007. Refinement of Building Facades by Integrated Processing of LIDAR and Image Data. In Proceedings of Photogrammetric Image Analysis (PIA07), Munich, Germany, 7-12.

[19] Wenzel, S., Drauschke, M., and Förstner, W. 2008. Detection of Repeated Structures in Facade Images. In Pattern Recognition and Image Analysis, Vol. 18, No. 3, 406-411.

[20] Früh, C., and Zakhor, A. 2004. An Automated Method for Large-Scale, Ground-Based City Model Acquisition. In International Journal of Computer Vision, 60 (1), 5-24.

[21] Cornelis, N., Cornelis, K., and Van Gool, L. 2006. Fast Compact City Modeling for Navigation Pre-Visualization. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Vol. 2, 1339-1344.

[22] Pollefeys, M., Nistér, D., Frahm, J.-M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.-J., Merell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G., and Towles, H. 2008. Detailed Real-Time Urban 3D Reconstruction from Video. In International Journal of Computer Vision, Vol. 78 (2-3), 143-167.

[23] Sinha, S.N., Frahm, J.-M., Pollefeys, M., and Genc, Y. 2006. GPU-Based Video Feature Tracking and Matching. In Proceedings of EDGE 2006, Workshop on Edge Computing using New Commodity Architectures, Chapel Hill, USA.

[24] Döllner, J., and Kyprianidis, J.E. 2009. Approaches to Image Abstraction for Photorealistic Depictions of Virtual 3D Models. Proceedings of the First ICA Symposium for Central and Eastern Europe, 371-385.

[25] Forberg, A. 2004. Generalization of 3D Building Data based on a Scale-Space Approach. In The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Istanbul, Turkey, Vol. XXXV, Part B.

[26] Thiemann, F., and Sester, M. 2004. Segmentation of Buildings for 3D-Generalisation. In Working Paper of the ICA Workshop on Generalisation and Multiple Representation, Leicester, UK.

[27] Poupeau, B., and Ruas, A. 2007. A Crystallographics Approach to Simplify 3D Building. In Proceedings of the 23rd XXIII International Cartographic Conference, Moscow, Russia.

[28] Kada, M. 2005. 3D Building Generalisation. In Proceedings of the 22th International Cartographic Conference, La Coruna, Spain.

[29] Thiemann, F., and Sester, M. 2006. 3D-Symbolization using Adaptive Templates. In Proceedings of the GICON 2006, Vienna.

[30] Kada, M. 2007. Scale-Dependent Simplification of 3D Building Models Based on Cell Decomposition and Primitive Instancing. In Spatial Information Theory: Proceedings of the 8th International Conference, COSIT 2007, 222-237.

[31] Sester, M. 2000. Maßstabsabhängige Darstellung in digitalen räumlichen Datenbeständen (German) (Postdoctoral thesis). Deutsche Geodätische Komission, Reihe C, Heft 544.

[32] Anders, K.-H. 2005. Level of Detail Generation of 3D Building Groups by Aggregation and Typification. In Proceedings of the 22th International Cartographic Conference, La Coruna, Spain.

[33] Glander, T., and Döllner, J. 2008. Automated Cell Based Generalization of Virtual 3D City Models with Dynamic Landmark Highlighting. In Proceedings of the 11th ICA Workshop on Generalization and Multiple Representation, Monpellier, France.

This work has been developed within the NEXUS project (Collaborative Research Centre 627 "Spatial World Models for Context-Aware Applications"), funded by the German Research Foundation (DFG).

Research Issues in Data Provenance for Streaming Environments

Position Paper

Hyo-Sang Lim†, Yang-Sae Moon‡, Elisa Bertino†
†*CERIAS and Department of Computer Science, Purdue University, USA*
{hslim, bertino}@cs.purdue.edu
‡*Department of Computer Science, Kangwon National University, South Korea*
ysmoon@kangwon.ac.kr

ABSTRACT

In this paper, we discuss research issues concerning data provenance for streaming environments. In data streams, especially in sensor networks, data provenance is a key information for assessing data quality since it gives important evidence about the origin of the data. We first show our initial approach for assessing trust scores of streaming data based on provenance. We then discuss open researches issues about using and delivering provenance in data streams.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Management, Measurement, Security

Keywords

Data Provenance, Data Stream, Security, Trustworthiness

1. INTRODUCTION

Pushed by recent advances in areas such as sensor networks, embedded systems, and ubiquitous/mobile computing, streaming management environments are being adopted in many applications. Such applications include traffic control systems monitoring data from mobile sensors, location based services (LBSs) based on user's continuously changing location, e-healthcare systems monitoring patient medical conditions, and realtime financial analysis. Sensor networks, which are a typical example of data stream environments, are also widely deployed in many different application domains from monitoring environments, such as pollution, temperature, and earthquake monitoring, to controlling automated systems, such as manufacturing facilities and power plants.

In these applications, data provenance is crucial for assessing data quality since data is originated by multiple sources and is processed by multiple intermediate agents. Here, data provenance refers to information documenting how data came to be in its current state - where it originated, how it was generated, and the manipulations it underwent since its creation. The location of sensors and nodes is a crucial component of provenance in that it may affect the quality of data. Data and service mashups accelerate the importance of data provenance since these applications integrate data from various sources and provide the combined data to users.

The use of provenance in data streams is relevant in many real applications. Two application examples are the following ones:

- A battlefield monitoring system gathers enemy locations from various sensors deployed in vehicles, aircrafts, and satellites and processes the monitoring queries over these streaming data. In this system, we need to assess the collected data since we must make sure that mission critical applications only access highly trustworthy data in order to guarantee accurate decisions by these applications. Since sensors and communication lines have different accuracy and confidence, it is essential to know the provenance of each data for assessing its trustworthiness level.
- A Supervisory Control And Data Acquisition (SCADA) system collects real-time information from data collection points such as sensors, and, based on analyses performed by a management system, performs process control tasks and monitors equipments from remote locations [8]. In this system, we must use only highly trustworthy data in order to prevent critical damages due to wrong control decisions because of wrong data. For example, in an electric power grid which consists of 270 utilities using a SCADA system that can contain up to 50,000 data collection points and over 3,000 public/private electric utilities, any single point of failure can disrupt the entire process flow and can potentially cause a domino effect that shuts down the entire systems [8]. As mentioned for the battlefield monitoring system, data provenance is a key information to prevent such destructive accidents.

ACM SPRINGL '09, November 3, 2009. Seattle, WA, USA

These examples show the importance of data provenance from the viewpoint of assessing trustworthiness of data items. For the sake of simplifying the presentation, in the rest of the paper we will focus on the use of provenance for assessing trust even though there are many other usages for provenance such as identifying/clustering data, comparing information from various sources, and detecting injections of malicious data.

Nevertheless the importance of data provenance, there are lots of challenges concerning the use of information in data streams due to the unique nature of the environment. The primary characteristic of data streams is that they arrive rapidly and should be processed in real-time [10, 14]. In real applications, data stream management systems (*DSMSs* for short) consume hundreds or thousands of data items per minute and process a large number of continuous queries which are registered in advance and run repeatedly whenever a new data item arrives [1, 14]. Therefore, high performance processing of data provenance is essential in data streams. Data streams are also potentially unbounded in size. This means that it is not possible to store the entire set of data items [14]. Therefore, we can only sequentially access data provenance and cannot refer to the past data provenance.

The size of data provenance also complicates its management in that it tends to increase as it is transmitted from the source to the server. The reason is that data provenance must also includes annotations concerning the intermediate processes performed on the data items. Such increasing size can slow down the network since the network bandwidth is usually limited in data stream environments, especially in wireless sensor networks.

Only a few research efforts have been reported concerning data provenance in data streaming. The first one is the IBM T.J. Watson's Century [3, 13], which is a biomedical data stream system for online healthcare analytics. Century provides a framework for analysis of sensor-based medical data, in which provenance plays a central role [13]. The main goal of the system is to support scalable automated near-real time analysis of high volumes of medical sensors. The system uses both data provenance and process provenance to describe which data and operators contributed to the generation of a particular data item. Century also supports historical data reply which recreates the processing graph that created the output.

Recently, Vijayakumar and Plale [18] have proposed a system architecture for near-real time provenance collection in data streams. They focus on identifying information which represents provenance of a data item from real time data streams, capturing provenance history of streams, tracing the source of a stream long after the process has completed. The system has been proposed for a specific application domain, that is, meteorology forecasting.

In this paper, we discuss research issues concerning data provenance in streaming environments. Specifically, we focus on two issues: 1) using data provenance in data streams and 2) delivering data provenance via low-bandwidth and insecure networks. We start with the first issue by presenting an overview of our initial solutions for assessing trust scores of streaming data based on provenance. In describing our solution, we first present the novel notion of confidence policy which controls the use of data based on the purpose of use and trust scores of the data, and then present the overall framework for enforcing the confidence policy in DSMSs. Then, based on our solution, we discuss the open research challenges.

The rest of the paper is organized as follows. Section 2 briefly presents our initial approach for assessing trust scores of streaming data in DSMSs. Section 3 discusses open researches issues related with data provenance in data streams. We finally summarize and conclude the paper in Section 4.

2. PROVENANCE-BASED CONFIDENCE POLICY CONTROL IN DATA STREAMS

In this section, we briefly survey our initial solution for enforcing confidence policies in data streams which is addressed in the previous work [12]. We focus on sensor networks since they are typical examples of data streams and have many interesting features related with provenance such as network topology and in-network processing issues.

We propose the novel notion of *confidence policy* that specifies the minimum trust score that a data item, or set of data items, must have for use by an application or task. Here, the trust score is associated with each data item and provides an indication of the trustworthiness of the data item. Confidence policies are integrated with query processing in that query results are filtered by the policies before being returned to the application. Our approach is based on the concept of provenance since more trustworthy sources generate more trustworthy data.

We first introduce a basic provenance model for sensor networks, and then, describe a DSMS-based framework for confidence policy management. We then introduce a systematic approach for assessing trust scores.

2.1 Data provenance model

We model the sensor network as a graph of $G(N, E)$ where N is a set of nodes (i.e., sensors) and E is a set of network connections between nodes. We assume that the sensors are in ad-hoc networks which send data items to DSMSs by relaying due to the insufficient capability of data transmission. In this kind of network, the sensors are categorized into three types according to their roles: 1) a *terminal node* generates a data item and sends it to one or more intermediate or server nodes; 2) an *intermediate node* receives data items from one or more terminal or intermediate nodes, and passes them to intermediate or server nodes; it may also generate an aggregated data item from the received data items and send the aggregated item to intermediate or server nodes; 3) a *server node* receives data items and evaluates continuous queries based on those items.

We introduce two types of data provenance: the *physical provenance* and the *logical provenance*. The physical provenance of a data item shows where the item was produced and how it was delivered to the server. We exploit the physical provenance to compute trust scores. The physical provenance can be represented as a path from a terminal node to a server node or a tree if there are more than two terminal nodes involved in the generation of the data item. The logical provenance of a data item represents the semantic meaning of the data item in the context of a given application. For example, the logical provenance can be a chain of employees who used the data item, or a trail of business logics that processed the data item. The logical provenance is used for grouping data items into semantic events with the same meaning or purpose.

2.2 Confidence policy control

Figure 1 shows our overall framework for confidence policy management. The framework shows how sensor data are processed and managed by a DSMS and how they are delivered to users. As shown in the figure, the proposed framework consists of three major components: *trust score computation*, *query and policy evaluation*, and *data quality management*. The role of each component is as follows:

- The *trust score computation* obtains trust scores of data items based on those of network nodes and (periodically) updates the trust scores to reflect the effect of the newly arrived data items. It also maintains and updates the trust scores of network nodes based on the scores of data items.
- The *query and policy evaluation* executes continuous queries, each of which has its own *confidence range*. We assume that each continuous query Q is given with its confidence range $[q_{max}, q_{min}]$. For each Q with $[q_{max}, q_{min}]$, this component first obtains the resulting data items by evaluating the given query Q and then returns the data items for which the trust scores are in the range $[q_{max}, q_{min}]$.
- The *data quality management* tries to control the data quality (manually or automatically) by adjusting data rates, increasing/decreasing the number of sensor nodes, or changing delivery paths. Obviously, the data quality affects the trust scores, and many approaches [7, 11, 15] in the context of sensor networks have addressed the issue of controlling data quality.

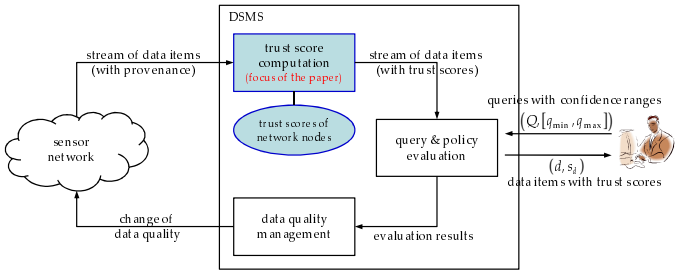


Figure 1: The overall framework for confidence policy control over data streams.

2.3 Computing trust scores

In data stream environments 1) data elements arrive incrementally and 2) trustworthiness of sensors can dynamically change as time goes on. Therefore, to provide accurate confidence information, we need a new framework that supports incremental assignment of trust scores for nodes and data items whose situations dynamically change.

To obtain trust scores, we propose a cyclic framework based on the *interdependency* [2, 6] between data items and their related network nodes. The interdependency means that the trust scores of data items affect the trust scores of network nodes, and similarly the trust scores of network nodes affect those of data items. In addition, the trust scores need to be continuously evolved in the stream environment since new data items continuously arrive to the server. Thus, a cyclic framework is adequate to reflect these interdependency and continuous evolution properties. Figure 2 shows the cyclic framework according to which the trust score of data items and the trust score of network nodes are continuously updated. The trust scores are computed for the data items of the same event (identified by logical provenance) in a given streaming window.

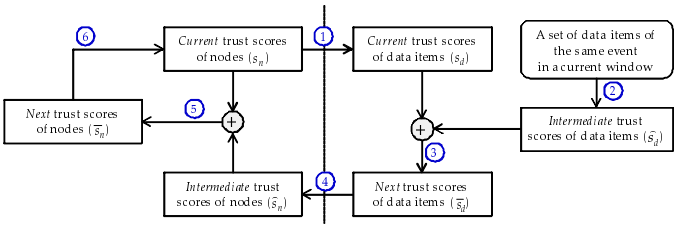


Figure 2: The cyclic framework for computing the trust scores of data items and network nodes.

As shown in Figure 2, we maintain three different types of trust scores, that is, *current*, *intermediate*, and *next trust scores*, to reflect the interdependency and continuous evolution properties in computing trust scores. We note that, since new data items are continuously added to the stream, executing the cycle once whenever a new data item arrives is enough to reflect the interdependency and continuous evolution properties in the stream environment.

Our framework works as follows. Trust scores are initially computed based on the values and provenance of data items; we refer to these trust scores as *implicit trust scores*. To obtain these trust scores, we use two types of similarity functions: *value similarity* inferred from data values, and *provenance similarity* inferred from physical provenances. Value similarity is based on the principle that the more data items referring to the same real-world event have similar values, the higher the trust scores of these items are. We observe that most sensor data referring to the same event follow the *normal distribution*, and propose a systematic approach for computing trust scores based on value similarity under the normal distribution. Provenance similarity is based on the observation that different physical provenances of similar data values may increase the trustworthiness of data items. In other words, different physical provenances provide more independent data items.

To the best of our knowledge, our approach is the first supporting the enforcement of confidence policies in DSMSs. However, there are still lots of open issues to be addressed. We discuss them in the next section.

3. OPEN RESEARCH ISSUES

We categorize open research issues into two categories: using data provenance and delivering data provenance.

3.1 Using data provenance

As we have seen in Section 2, data provenance can be used to guarantee the quality of results by enhancing the knowledge about data quality such as accuracy, confidence, and trustworthiness. Here, the most interesting issues are 1) how to exactly get the knowledge from data provenance and 2) how to efficiently use data provenance in data stream processing. In this discussion, we point out the differences between static databases and dynamic data streams.

3.1.1 Measuring data quality

Data provenance representation. In our initial approach [12], we use a simple representation of data provenance: a path from a terminal node to a server node or a tree with two or more terminal nodes. However, this simple model is not enough for complex applications which have more than two outgoing directions (e.g., a data item can be

transmitted to more than two intermediate nodes) or loops (e.g., a same process can be repeatedly conducted on a data item) in data flows. Because this kind of complex provenance often arises in applications such as document workflow systems and manufacturing processes, we need to consider a more general representation for both physical and logical provenances.

Provenance similarity. The provenance representation also affects the similarity measure between provenances. One of the popular similarity measures in graph theory is the edit distance which uses the minimum amount of distortion that is needed to transform one graph into another [4]. We also use a variety of edit distance since it well captures similarity or dissimilarity for both physical and logical provenances. However, computing the edit distance for general graphs is known to be an NP-hard problem [9]. Therefore, we need an approximate similarity measure method which efficiently compares graph-shape provenances for streaming data whose input rate is very high. If the number of possible provenance patterns is limited and small, pre-calculating the similarity among all possible provenances can be a feasible approach to reduce the runtime latency for calculating similarity.

Multiple data provenances. Even though we only use a single physical provenance and logical provenance [12], there can be multiple provenances in many real applications. For example, in the battlefield monitoring system mentioned in Section 1, the physical provenance can be a GPS location of sensors and an ID of the vehicle where the sensor is deployed. Also, the logical provenance can be the unit name of the troops formation and a sensor category where the sensor is classified into. In this multiple provenance situation, trust score assessment becomes more complex since all these provenances should be considered together. For example, there can be a conflict between two scores calculated based on two different provenances. However, the availability of multiple provenances can improve the data precision and the robustness for calculating trust score.

Dynamic data provenance. In our previous work [12], we assumed that the provenance of a sensor does not change. However, in the battlefield monitoring example, we can see that the physical provenance of a sensor can change because of mobility. The GPS location of a sensor deployed in an aircraft continuously changes during the sensing process. It means that data items from the same sensor can have different physical provenance. To handle such dynamic provenance, we need to combine variable provenance (e.g., GPS location) with fixed provenance (e.g., sensor ID) and use their relationship to calculate trust scores.

Quality measurement model. In our initial solution [12], we use a simple model for calculating trust scores: a normal distribution to model data similarity, a weighted sum to calculate data trust scores from node trust scores, and a simple method to reflect provenance similarity into the scores. However, if the application becomes more complex, this simple model cannot yield accurate trust scores. For example, sensed data may not follow the normal distribution, and choosing weights in the weighted-sum may not be possible in some applications. Therefore, we need to investigate more sophisticated and accurate quality measurement models with real applications.

3.1.2 Combining data provenance with query processing

Improving performance. Addressing the problem of efficiently computing trust scores is essential in data streams due to fast input rates and realtime processing requirement. In principle, trust scores should be evaluated whenever a new data item arrives. However, such a strategy is not applicable when input rates are very high. To address this problem, we need to develop methods to reduce the number of evaluations without sacrificing accuracy of trust scores. A possibility is to use a batch method which recalculates trust scores only when recent input data items do not follow current data distributions any longer.

Improving data quality. We can use data provenance for not only measuring but also improving data quality. As we discussed in Section 2, the data quality management component tries to improve data quality when the number of results satisfying a confidence policy is too small. In data stream environments, we can systematically improve data quality with provenance information. Along with our quality measurement model, we can more clearly identify which data from which sensors should be improved to increase the trust scores of results.

3.2 Delivering data provenance

Until now, we have assumed that the correct data provenance arrives with stream data and have focused on how to use them for stream processing. In this section, we focus on how to efficiently and securely deliver data provenance through networks.

3.2.1 Efficient delivery

Reducing the size of data provenance. Because network bandwidth is limited, we need to minimize the size of data provenance. The first method we can consider is eliminating redundant information. For example, if two contiguous data items have the same provenance, we can transmit the provenance only once for the two data items. If two contiguous data items have similar provenance (e.g., only differ with respect to the terminal nodes), we can only transmit the difference. For this purpose, we can exploit the concept of punctuation [17] which describes annotations for data items. Another method to reduce the size of data provenance is to use a bit-map representation with compression techniques. If the network structure is fixed during processing, we can use bit-maps to describe provenances in the network and use compression techniques to reduce the size of the representation.

Controlling the granularity of data provenance. If partial loss of provenance information can be tolerated, we can use the granularity of data provenance to reduce the amount of information. For example, in a geographical provenance, we can use state level provenance (i.e., coarse granularity) instead of city level provenance (i.e., fine granularity). Here, the city level source information can be omitted in the provenance and data items from the same state can share the provenance if the punctuation scheme is used. The provenance granularity has to be chosen according to the semantics of applications so that the loss of provenance semantics is minimized

3.2.2 Secure delivery

Avoiding unauthorized modification. Since data provenance is a key evidence for measuring the quality of data

items, it should be protected from malicious and unauthorized modification. For example, if the provenance of an untrustworthy data item is modified to a trustworthy node, the quality of the data item is assessed as high, even though in reality the data item has a low quality. However, it is not easy to protect data provenance from such attacks due to the nature of data streams, fast input rate and real-time requirement. Conventional digital signature and cryptography techniques cannot be used in fast data streams since their cost is too high compared with the cost of ordinary continuous query processing. We can reduce the cost by using these techniques only for data provenance and not for data values. However, we still need to develop new light-weighted digital signature techniques for data provenance in fast data stream environments.

Handling loss of data provenance. When using a punctuation scheme to describe data provenance, dropping punctuations can be a mean by an attacker to increase the trust scores of certain data items. For example, if a malicious user intentionally drops a punctuation which precedes data items that are from an untrustworthy source, the provenance of the data items can be misinterpreted by other punctuations which are not related with them. To avoid this kind of attack, one approach is to use digital watermark techniques which embed provenance inside the data. Since digital watermark techniques provide robustness against dropping partial information, data provenance can be sustained even when some data provenance information (or punctuation) is omitted.

4. CONCLUSION

In this paper, we have discussed open research issues concerning data provenance in data stream environments. We have focused on two issues: 1) using data provenance in data streams and 2) delivering data provenance in low-bandwidth and insecure networks. We believe that the relevance of research in this area will become higher with the the growth of data stream applications and ubiquitous computing environments.

Acknowledgement. This work has been partially supported by AFOSR grant FA9550-07-1-0041 "Systematic Control and Management of Data Integrity, Quality and Provenance for Command and Control Applications".

5. REFERENCES

[1] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and Issues in Data Stream Systems," In *Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, Madison, WI, pp. 1-16, June 2002.

[2] E. Bertino, C. Dai, H.-S. Lim, and D. Lin, "High-Assurance Integrity Techniques for Databases," In *Proc. of the 25th British Nat'l Conf. on Databases*, Cardiff, UK, pp. 244-256, July 2008.

[3] M. Blount et al., "Century: Automated Aspects of Patient Care," In *Proc. of the 13th IEEE Int'l Conf. on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007)*, Daegu, Korea, pp.504-509, August 21-24, 2007.

[4] H. Bunke and G. Allermann, "Inexact graph matching for structural pattern recognition," *Pattern Recognition Letters*, vol. 1, pp.245-253, 1983.

[5] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu, "An Approach to Evaluate Data Trustworthiness Based on Data Provenance," In *Proc. of the 5th VLDB Workshop on Secure Data Management*, Auckland, New Zealand, pp. 82-98, Aug. 2008.

[6] C. Dai et al., "Query Processing Techniques for Compliance with Data Confidence Policies," In *Proc. of the 6th VLDB Workshop on Secure Data Management*, Lyon, France, pp. 49-67, 2009.

[7] Y. David, N. Erich, K. Jim, and S. Prashant, "Data Quality and Query Cost in Wireless Sensor Networks," In *Proc. of the 5th IEEE Int'l Conf. on Pervasive Computing and Communications Workshops*, White Plains, New York, USA, pp. 272-278, Mar, 2007.

[8] J. D. Fernandez and A. E. Fernandez, "SCADA Systems: Vulnerabilities and Remediation," *Journal of Computing Sciences in Colleges*, Vol. 20, No. 4, pp. 160-168, Apr. 2005.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1990.

[10] L. Golab and T. Ozsu, "Issues in Data Stream Management," *ACM SIGMOD Record*, Vol. 32, No. 2, pp. 5-14, June 2003.

[11] S.-Y. Hwang, P.-Y. Liu, and C.-H. Lee, "Using Resampling for Optimizing Continuous Queries in Wireless Sensor Networks," In *Proc. of the 8th Int'l Conf. on Intelligent Systems Design and Applications*, Kaohsiung, Taiwan, pp. 107-110, Nov. 2008.

[12] H.-S. Lim, Y.-S. Moon, and, E. Bertino, "Provenance-based Confidence Policy Management in Data Streams," under submission.

[13] A. Misra, M. Blount, A. Kementsietsidis, D. M. Sow, and M. Wang, "Advances and Challenges for Scalable Provenance in Stream Processing Systems," In *Proc. of the 2nd Int'l Provenance and Annotation Workshop (IPAW 2008)*, Salt Lake City, UT, USA, pp. 253-265, June 17-18, 2008.

[14] R. Motwani et al., "Query Processing, Approximation, and Resource Management in a Data Stream Management System," In *Proc. of the First Biennial Conf. on Innovative Data Systems Research*, Asiloma, California, pp. 245-256, Jan. 2003.

[15] C. Olston, J. Jiang, and J. Widom, "Adaptive Filters for Continuous Queries over Distributed Data Streams," In *Proc. of the Int'l Conf. on Management of Data*, ACM SIGMOD, San Diego, CA, pp. 563-574, June 2003.

[16] R. V. Nehme, E. A. Rundensteiner, and E. Bertino, "A Security Punctuation Framework for Enforcing Access Control on Streaming Data," In *Proc. of the 24th Int'l Conf. on Data Engineering*, Cancun, Mexico, pp. 406-415, Apr. 2008.

[17] P. A. Tucker, D. Maier, T. Sheard, and L. Fegaras, "Exploiting Punctuation Semantics in Continuous Data Streams," *IEEE Trans. on Knowledge Data Engineering*, Vol.15, No.3 pp. 555-568, May 2003.

[18] N. Vijayakumar and B. Plale, "Towards Low Overhead Provenance Tracking in Near Real-Time Stream Filtering," In *Proc. of the Int'l Provenance and Annotation Workshop (IPAW 2006)*, Chicago, USA, 2006.

On the Impact of Localization Data in Wireless Sensor Networks with Malicious Nodes

Mattia Monga
Dip. di Informatica e Comunicazione
Università degli Studi di Milano
Via Comelico 39, I–20135 Milano, Italy
mattia.monga@unimi.it

Sabrina Sicari
Dip. di Informatica e Comunicazione
Università degli Studi dell'Insubria
Via Mazzini 5, I–21100 Varese, Italy
sabrina.sicari@uninsubria.it

ABSTRACT

The knowledge of node positions is a core concept in any Wireless Sensor Network context. Several localization algorithms were devised, but secure localization of sensor nodes is still a challenging task to achieve with a high level of performance. In fact, location information might be the target of different kinds of malicious attacks and several secure localization approaches were proposed. In this paper we analyze the impact of false data in a secure localization algorithm, known as *Verifiable Multilateration*. We found that the strategy used to compute the positions of nodes might have an impact both on the computational effort needed to achieve acceptable measures and the precision of the detection of malicious nodes.

Categories and Subject Descriptors

K.6.5 [**MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS**]: Security and Protection

General Terms

Wireless Sensor Networks, security

Keywords

WSN, security, localization

1. INTRODUCTION

Several researchers are proposing information systems based Wireless Sensor Networks (WSNs), that provide a flexible and effective means to monitor large and diverse geographical areas. However, WSNs are composed by individual nodes with very limited capabilities and energy consumption is a major concern, thus unorthodox solutions are required for many situations, especially aimed at minimizing the communication overload. Moreover, the monitoring activity greatly relies on data about the positions of nodes, which are often deployed randomly, thus a great challenge is represented by localization at time of operations [12].

Various location services have been proposed. The Global Positioning System (GPS) is the most well-known location service in use today, but it is unsuitable for low-cost, ad-hoc sensor networks since GPS is based on an extensive infrastructure (i.e. satellites) that requires frequent transmissions and devices are still quite expensive and heavy. Likewise the solutions developed in the area of robotics [1, 13, 24] and ubiquitous computing [10] are generally not applicable for sensor networks as they require too much processing power and energy. Recently a number of localization systems have been proposed specifically for sensor networks [3, 4, 7, 9, 23, 16, 19, 22]. Ideally, these approaches aim at large-scale ad-hoc sensor networks (100+ nodes) and their design goals are:

- to be as much as possible self-organizing, thus that communication happens mostly locally, without the need of a globally accessible infrastructure;
- to be tolerant to node failures and range errors;
- to require little computation and, especially, communication effort.

Unfortunately, most of the current approaches omit to consider that WSNs could be deployed in an adversarial setting, where hostile nodes under the control of an attacker coexist with faithful ones. In fact, from a security point of view, the wireless communications and the deployment in uncontrolled environments rise several issues: the confidentiality, the integrity, and the availability of data might be put at risk by malicious tampering of sensors and/or traffic.

Node position is a really critical information due to the strict relation with the quality of the provided services. In fact, the location information is sometimes target of different kinds of malicious attacks, classified in internal and external attacks. So the trustworthiness of node position information is a challenging task for wireless sensor networks since classical solutions based on access control and strong authentication, are problematic to implement with limited resources and short battery life. Also, nodes are prone to physical attacks and is pretty easy to clone a sensor device and its on-board keys: thus cryptography provides only a partial protection and should be used with care.

In this paper we analyzed an approach to the secure localization of nodes known as *Verifiable Multilateration* (VM) [5]. VM potentially uses untrusted information to derive the positions of nodes, together with a measure of their trustworthiness. VM relies on *lateration* to compute positions, a

ACM SPRINGL'09 November 3, 2009. Seattle, WA, USA

generalization of triangulation to multiple nodes: several techniques to do lateration are known and VM is largely independent from the choice of one of them. We found, however, that the choice is not neutral. Lateration algorithms can be computationally heavy in order to get very precise results or very trivial if only gross data are needed. One could legitimately ask if and when the additional efforts are needed and at what level, since computation in WSNs is a scarce resource. In this paper we try to answer to these questions by analyzing three lateration algorithms and assessing their impact on VM. We will show that in general the precision is worth the computational price, but, under some hypotheses, even a trivial solution can be acceptable.

The paper is organized as follows: Section 2 provides a short state of art about the sensor node localization solutions; Section 3 describes the reference scenario in which we performed our analysis; Section 4 introduces the Verifiable Multilateration for secure localization; Section 5 analyzes three different approaches to lateration and their impact on secure localization and finally, Section 6 draws some conclusions and provides hints for future works.

2. RELATED WORK

All the proposed localization algorithms for wireless sensor networks have to face the particular context in which sensor nodes are deployed. More specifically, there is in general no fine control over the placement of the sensor nodes when the network is deployed (e.g., when nodes are dropped from an airplane) and some self organization of the communication overlay is needed. Moreover, the connectivity of the nodes in the network (i.e., the average number of neighbors) is another important parameter that has a strong impact on the accuracy of most localization algorithms. In fact, the main node position approach is based on node density and radio range, and in some cases it can be dynamically adjusted by changing the transmit power of the RF radio. So taking into account the domain constraints, any localization algorithm has to address three main requirements: self organizing, robustness and energy efficiency.

Existing localization schemes may be classified in *range-based methods*, which use exact measurements of distances, and *range-free methods*, which only need beacon signals. Typical techniques to obtain the measurements between two nodes include Received Signal Strength Indicator (RSSI), Time of Arrive (ToA), Time Difference of Arrive (TDoA), and Angle of Arrive (AoA). Range-based localization schemes in sensor networks include those in [21, 22, 17, 15, 9]. Savvides et al. developed an ad-hoc localization system localization protocol based on TDoA [21]. Extension of this work can be found in [22]. Doherty et al. presented a localization scheme based on connectivity induced constraints and the relative angle between neighbors [9]. AoA is also used to develop localization schemes in [17] and [15]. Range-free based schemes are proposed to provide location estimation services for those applications with less required location precision [3, 14, 23]. Estrin et al. proposed a simple range-free, coarse grained localization scheme where each sensor estimated its location by centering the locations contained in the received signals [3]. All of the current localization schemes become vulnerable when there are malicious attacks. In all these schemes, the accuracy of location estimation depends on the accuracy of the origins of the beacon signals and certain measurements obtained from the beacon signals, including distances and/or angles in range-based schemes, and the existence of beacon signals in range-free schemes. Though the above measurements are directly obtained from the physical signals, the locations of the beacon signals' origins can be easily forged. As a result, a malicious attacker may introduce large errors when a node estimates its location. More specifically, an attacker can introduce arbitrarily large errors by declaring false locations in beacon packets, arbitrarily introducing large errors into a non-beacon node's location estimation. Such attacks cannot be simply prevented by cryptographic techniques due to the threat of compromised nodes and replay attacks. In order to overcome such a limit localization algorithms adopt some techniques able to reveal malicious behavior.

We focus our attention on the secure localization algorithm, named Verifiable Multilateration (VM) [5] that is a range based approach using MMSE as one criterium for revealing malicious behavior. Our choice of MMSE is due to the robustness towards attacks and the capability to reveal malicious behavior and then the accuracy of the obtained results as we show in details in the following sections. We aim at analyzing in some depth the computational effort of minimization, by comparing MMSE with other available solutions. Since a weak assessment of localization information may damage service performance, our goal is to understand the trade-off between the computational cost and the overall trustworthiness of the obtained results.

3. REFERENCE SCENARIO

We consider a dense network composed of nodes n_i, where $n_i \in N, 0 < i \leq |N|$ and a base station b in which all the collected data sink. We consider two subsets of N:

- S, composed by nodes $s_i, 0 < i \leq |S|$, which perform sensing functions;
- V, composed by nodes $v_i, 0 < i \leq |V|$, which work as verifiers in the secure localization protocol.

$N = S \cup V$ and V may overlap S (in principle every node whose position can be taken for granted might be used as a verifier).

Each s_i node senses a given type of data (*e.g.,* temperature, pressure, brightness, position and so on). Each node (sensing, and verifier) directly communicates with its closer neighbours (at one hop distance).

4. SECURE LOCALIZATION

The node positions can be evaluated by using a multilateration technique, which determines the node coordinates by exploiting a set of landmark nodes, called *anchor* nodes, whose positions are known. The position of the unknown node u is computed by using an estimation of the distances between the anchor nodes and the node itself. The distance is not measured directly; instead, it can be computed by knowing the speed of the signal in the medium used in the transmission, and by measuring the time needed to get an answer to a beacon message sent to u. If the computation is carried on without any precaution, u might fool the anchors by delaying the beacon message. However, since a malicious node can delay the answer beacon, but not speed it up, under some conditions it is possible to spot malicious behaviors. *Verifiable Multilateration* (VM) [5] uses three or

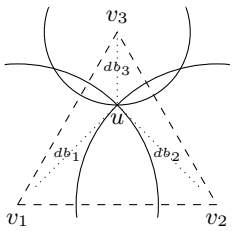


Figure 1: Verifiable multilateration

more anchor nodes to detect misbehaving nodes. In VM the anchor nodes work as *verifiers* of the localization data and they send to the sink b the information needed to evaluate the consistency of the coordinates computed for u. The basic idea of VM is shown in Figure 1: each verifier v_i computes its *distance bound* [2] to u; any point $u' \neq u$ inside the triangle formed by v_1, v_2, v_3 has necessarily at least one of the distance to the v_i enlarged. This enlargement, however, cannot be masked by u by sending a faster message to the corresponding verifier. Therefore, if the verifiers are trusted and they can securely communicate with b, the following algorithm can be used to check the localization data:

1. Each verifier v_i sends a beacon message to u and records the time τ_i needed to get an answer;
2. Each verifier v_i (whose coordinates $< x_i, y_i >$ are known) sends to b a message with its τ_i;
3. From τ_i, b derives the corresponding distance bound db_i (that can be easily computed if the speed of the signal is known) and it estimates u's coordinates by minimizing the mean square error

   $$\epsilon = \sum_i (db_i - \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2})^2$$

   where $< x_u, y_u >$ are the (unknown) coordinates to be estimated[1].;
4. b can now check if $< x_u, y_u >$ are feasible in the given setting by two incremental tests:
   (a) *δ-test:* For all verifiers v_i, compute the distance between the estimated u and v_i: if it differs from the measured distance bound by more than the expected distance measurement error, the estimation is affected by malicious tampering;
   (b) *Point in the triangle test:* Distance bounds are reliable only if the estimated u is within at least one verification triangle formed by a triplet of verifiers, otherwise the estimation is considered unverified.

If both the δ and the point-in-the-triangle tests are positive, the distance bounds are consistent with the estimated node position, which moreover falls in at least one verification triangle. Thus, the sink can consider the estimated position of the node as `Robust`; else, the information at hands is not sufficient to support the reliability of the data. An estimation that does not pass the δ test is considered `Malicious`. A sensible value of δ depends on the expected error in time measurement and the number of available verifiers. The simulation reported below should clarify the considerations involved in the choice of δ. If the δ test is passed, but the point-in-the-triangle one fails, the sink marks the estimation as `Unknown`, meaning there is no sufficient information for evaluating the trustworthiness of node position. Thus, the localization phase ends up, for each unlocalized node u_i, with an estimation of the position of u_i and a quality $W_i \in Robust, Unknown, Malicious$.

[1]In an ideal situation where there are no measurement errors and/or malicious delays this is equivalent to finding the (unique) intersection of the circles defined by the distance bounds and centered in the v_i (see Figure 1) and $\epsilon = 0$

5. THE IMPACT OF LOCALIZATION INFORMATION

Summing up, VM aims at assessing the trustworthiness of a node position by checking the consistency of the data received by the sink:

- the δ test establishes a threshold incompatible with highly deceptive data;
- the point-in-the-triangle test rules out geometrically infeasible deceptions.

As stated above, the original VM approach requires (step 3) the minimization of the mean square error ϵ. This function, however, is not linear and minimization is far from trivial. In fact, no exact solution is possible and some approximation is needed. In our experiments, we found that most of the computational effort of the approach was in the minimization. Thus, we considered three alternatives:

1. use a probabilistic heuristic to approximate the search for the minimum ϵ (MMSE approach)
2. use a function easier to deal with (exact lateration approach)
3. use a trivial estimate of the position (min-max approach)

In order to analyze the feasibility of these simplifications in an adversarial context, we used OMNeT++ (ver. 3.3p1, [8, 18]) to set up a simulation of the secure localization algorithm. A claimant node u to be localized resides at the center of a 100m×100m field, *i.e.,* at point $< 50, 50 >$. Since the best approach to lay out three verifiers is on the vertexes of an equilateral triangle [5], we fixed their coordinates to be the points $< 1, 1 >, < 99, 1 >, < 50, 85 >$. If u is *faithful,* it answers to verifiers' beacons without any delay. Otherwise, if u is *malicious* it adds a variable delay to the answers, in order to dissimulate a fake position u': *i.e.,* for each v_i, if the distance $v_i\bar{}u'$ is greater than $v_i\bar{}u$ a proper delay is added by u to the answer beacon to v_i. We assumed that signals travel at the speed of light and that time can be measured with an error whose standard deviation is 2ns. As described above, the timing information collected by verifiers v_i can be used by the base station to classify the claimant as `Malicious`, `Unknown`, or `Robust`.

In a preliminary study, we discovered that the error introduced by the localization heuristic is indeed critical, since it could cause an unexpected behavior in the algorithm. Figure 2 shows a number of anchors v_i and the distance bounds they estimate in color. The actual position of the malicious

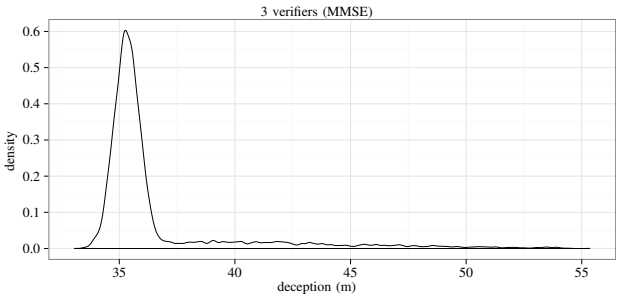


Figure 4: Deception when a node is classified as `Unknown` ($\delta = 35$)

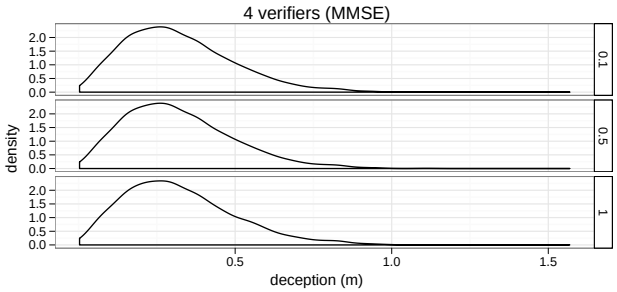


Figure 5: Deception when a malicious node is classified as `Robust`

claimant node is u and the estimated position u'. In Figure 2(a), with three verifiers, the node results as `Unknown` since it is outside the unique verification triangle. However, when a fourth verifier is added to the system (see Figure 2(b)) the estimation u' falls inside at least one of the four verification triangles and the node ends up to be `Robust`.

Thus, we decide to analyze the sensitivity of VM to the localization heuristic used. In particular, we considered as a quality metric the *deception* that can be induced by an attacker, *i.e.,* the distance between the actual node position and the estimated one, when node are classified as `Robust` or `Unknown`.

5.1 The MMSE approach

The original proposal of VM, relies on the minimization of the mean square error (MMSE). In our simulation, we used *simulated annealing* [11, 6] heuristic to approximate a solution.

Figure 3(a) shows the effect of the choice of the δ_{max} in the δ-test on 10000 runs with 3 verifiers: the only sensible value is about 35, since lower levels have an overwhelming rate of false positives (*i.e.,* faithful nodes classified as `Malicious`), and a higher δ gives too much false negatives (*i.e.,* malicious nodes classified as `Robust`) and unknowns. About 50% of malicious claimants and 90% of faithful ones were classified as `Unknown`: the error in taking the estimated position instead of the real one is pretty high, as one can see from Figure 4 that plots the density of deception: most of the time by accepting an estimation classified as `Unknown` one has to deal with a deception of about 35 m. The situation is clearly improved when a fourth verifier is added (see Figure 3(b)): the setting is now with a verifier at each corner of the field and all the values less than 2.5 give acceptable results; there are no `Unknowns`. It is worth noting that the range of δ considered is different, since by increasing the number of verifiers, the maximum acceptable error δ_{max} should decrease. There are still some false negatives, but the deception induced by a malicious node taken as `Robust` is always less than 1m with $\delta \leq 1$. Figure 5 plots the density distribution of the deception — *i.e.,* the distance between the real position and the estimated one — at different values of δ. Adding a fifth verifier randomly deployed significantly decreases the rate of false negatives, as shown in Figure 3(c).

5.2 The exact lateration approach

An easier estimation to compute is *exact lateration* (used for example by [16, 20]) that considers the system of equations

$$\forall i, 1 \leq i \leq |V| : \quad (x_i - x_u)^2 + (y_i - y_u)^2 = db_i \qquad (1)$$

The system (1) can be linearized by subtracting the last equation (the one corresponding to verifier $|V|$) from the other $|V| - 1$ ones.

$$\forall i, i \neq |V| :$$
$$x_i^2 - x_{|V|}^2 - 2(x_i - x_{|V|})x_u + y_i^2 - y_{|V|}^2 - 2(y_i - y_{|V|})y_u = db_i^2 - d_{|V|}^2$$

The system above can be expressed in the matrix form

$$A_{(|V|-1,2)}x_{(2,1)} = b_{(|V|-1,1)}$$

where

$$A_{(|V|-1,2)} = [2(x_i - x_{|V|}) \quad 2(y_i - y_{|V|})]$$
$$b_{(|V|-1,1)} = [x_i^2 - x_{|V|}^2 + y_i^2 - y_{|V|}^2 + db_{|V|}^2 - d_i^2]$$

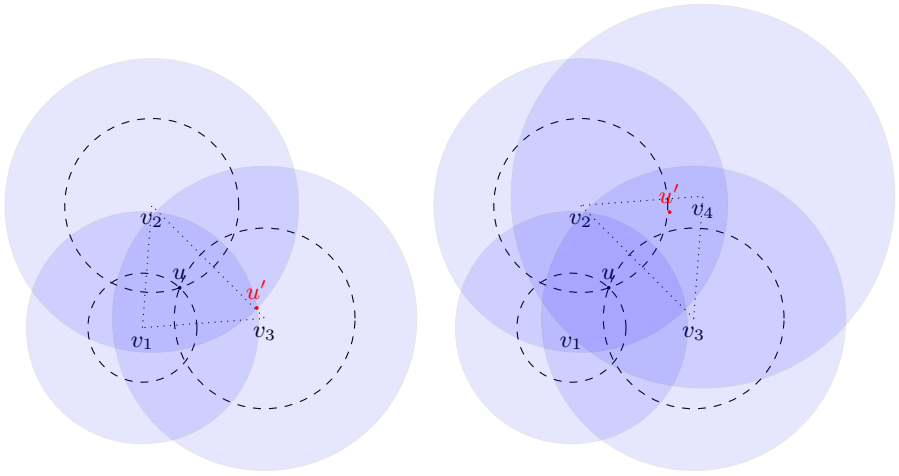
The system can be solved by using a standard linear algebra least-squares approach: $x = (A^T \cdot A)^{-1} \cdot A^T \cdot b$. A measure of the quality of the solution is then given by

$$r_{lat} = \frac{\sum_i (db_i - \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2})^2}{|V|}$$

In order to evaluate this approach to localization with respect to the MMSE one described in Section 5.1, we considered the relation between the residue r_{lat} (analogous to ϵ in the MMSE case) and the deception induced by assuming $< x_u, y_u >$ as the position. Figure 6 shows the correlation between the quality of the estimation and deception for both MMSE (Figure 6(a)) and exact lateration (Figure 6(b)): the latter is more spread, thus indicating that MMSE ϵ is a better proxy indicator for deception. In fact, deception by a malicious claimant evaluated by an exact lateration approach gives results fairly uncorrelated with the ones obtained with MMSE (see Figure 7).

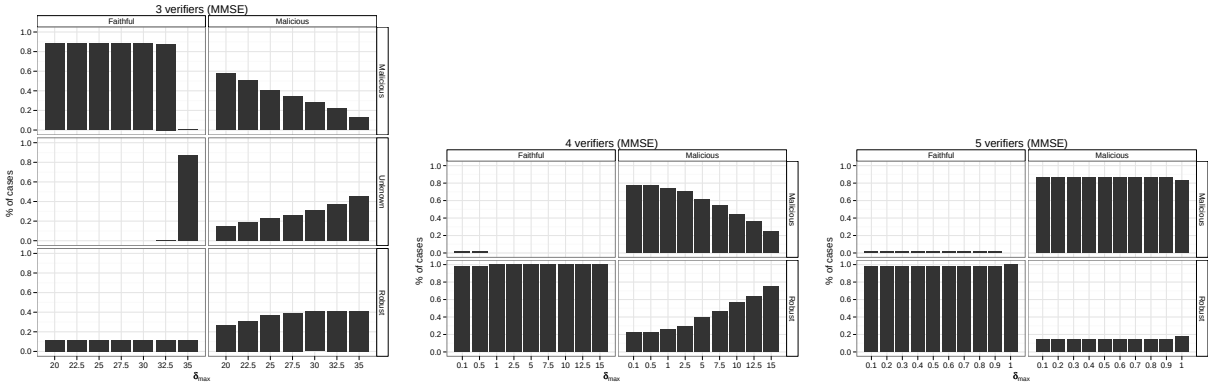
5.3 The min-max approach

Sometimes an even easier estimation used is the *min-max* method ([21], the name is coined in [12]). Its computation is almost trivial: for each verifier one considers the bounding box defined by $< x_i - db_i, y_i - db_i > -< x_i + db_i, y_i +$



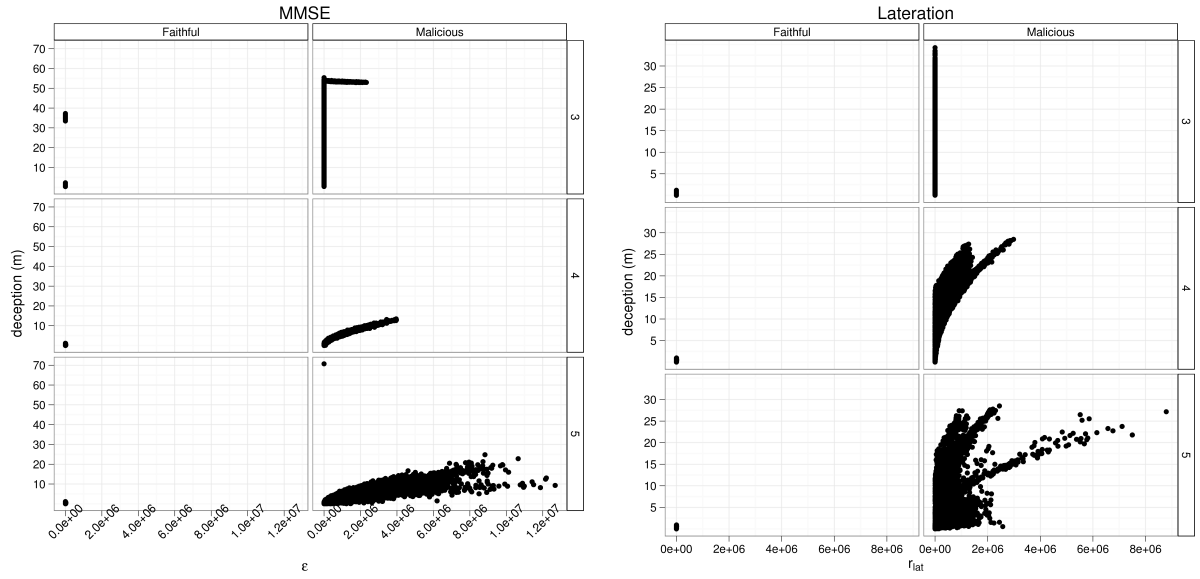
(a) (b)

Figure 2: The quality of the estimated position u' depends on the number of verifiers



(a) (b) (c)

Figure 3: Classification by secure localization



(a) (b)

Figure 6: Correlation between estimation quality and deception with 3,4, and 5 verifiers

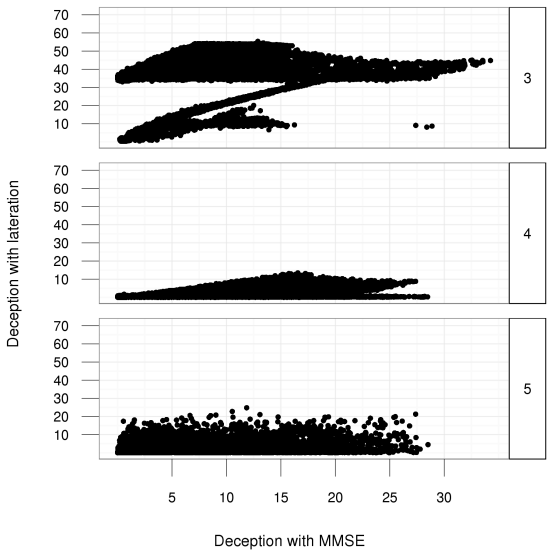


Figure 7: Deception by a malicious node evaluated by exact lateration with 3,4, and 5 verifiers

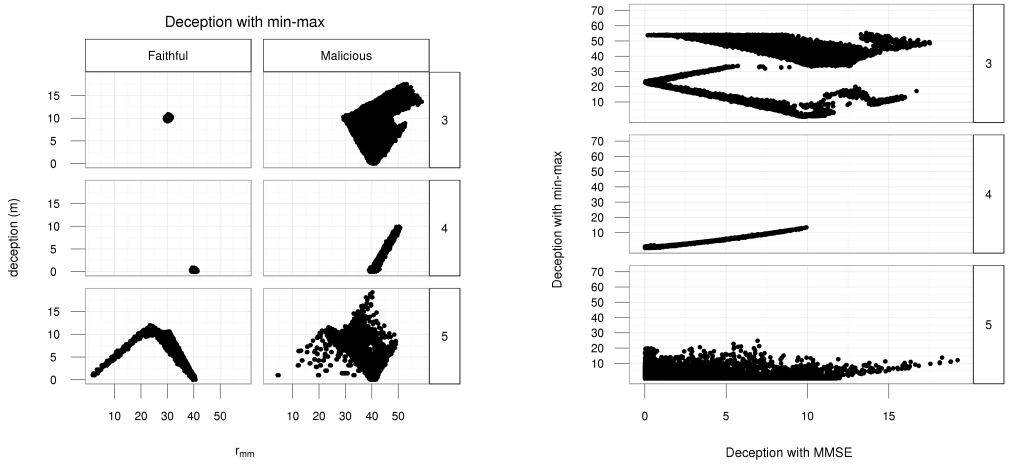
$db_i >$. The intersection of all bounding boxes is then computed as $< \max(x_i - db_i), \max(y_i - db_i) > - < \min(x_i + db_i), \min(y_i + db_i) >$ and the final position estimated is $\frac{<\max(x_i - db_i) + <\min(x_i + db_i)}{2}, \frac{<\max(y_i - db_i) + <\min(y_i + db_i)}{2} >$. We measured the quality of the estimation as

$$r_{mm} = |\frac{\max(x_i - db_i) - \min(x_i + db_i) + \max(y_i - db_i) - \min(y_i + db_i)}{2}|$$

Again, we found that MMSE ϵ is a much better proxy for deception in an adversarial setting (see Figure 8(a)). However, with four verifiers (posed on the vertexes of the rectangular field) the results would be consistent with the ones obtained via MMSE, but with a considerable saving in computation (see Figure 8(b)). However, this result is not confirmed in the 5-verifiers case: in fact, the fifth verifier — randomly deployed — destroys the symmetry of bounding boxes, and it has an unexpected detrimental effect. The setting with four verifiers, instead, could be a good alternative to the MMSE corresponding solution since it can give proportionally equivalent result with a much reduced computational effort.

6. CONCLUSIONS

Reliability of node positions is a core requirement in most Wireless Sensor Networks. Verifiable Multilateration uses potentially untrusted information to derive the positions of nodes, together with a measure of their trustworthiness. However, VM itself relies on node positions deduced by lateration. We analyzed different approaches to lateration in order to understand when the computational effort needed by the most sophisticated algorithms is really needed. Our results show that in general the precision provided by the most onerous algorithm (MMSE) is indeed needed. However, if a careful position of verifiers is possible, the much simpler min-max method could be useful. The aim of secure



(a) Correlation between estimation quality and deception

(b) Deception with a malicious node via MMSE and min-max

Figure 8: Estimation quality and deception in the min-max approach with 3,4, and 5 verifiers

localization algorithms is to define some criteria in order to identify and remove malicious positions. False positive may always occur, however, and one has to spend in verifiers and communication to increase the quality of the collected information. We are currently investigating the use of cross-layer information to assess the overall quality of the monitoring performed by the WSN and a game theoretical approach to model malicious behavior, in order to reason about the rational strategies open to the system designers.

Acknowledgment

This research has been partially funded by the European Commission, Programme IDEAS-ERC, Project 227977-SMScom.

7. REFERENCES

[1] S. Atiya and G. Hager. Real-time vision-based robot localization. *IEEE Trans. on Robotics and Automation*, 9(6):785–800, 1993.

[2] S. Brands and D. Chaum. Distance-bounding protocols. In *Proc. of Workshop on the Theory and Application of Cryptographic Techniques*, 1994.

[3] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, Oct. 2000.

[4] S. Čapkun, M. Hamdi, and J.-P. Hubaux. Gps-free positioning in mobile ad-hoc networks. *Cluster Computing*, 5(2):157–167, April 2002.

[5] S. Čapkun and J. Hubaux. Secure positioning in wireless networks. *IEEE Journal On Selected Areas In Communications*, 24(2):221–232, Feb. 2006.

[6] V. Cerny. A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–51, 1985.

[7] J. Chen, K. Yao, and R. Hudson. Source localization and beamforming. *IEEE Signal Processing Magazine*, 19(2):30–39, 2002.

[8] O. Community. `http://www.omnetpp.org/`.

[9] L. Doherty, K. Pister, and L. E. Ghaoui. Convex position estimation in wireless sensor networks. In *Proc. of IEEE Infocom 2001*, 2001.

[10] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, 2001.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, New Series*, 220(4598):671–680, May 1983.

[12] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Elsevier Computer Networks*, 43:499–518, 2003.

[13] J. Leonard and H. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Automation*, 7(3):376–382, 1991.

[14] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Proc. of IPSN'03*, 2003.

[15] A. Nasipuri and K. Li. A directionality based location discovery scheme for wireless sensor networks. In *Proc. of ACM WSNA'02*, 2002.

[16] D. Niculescu and B. Nath. Ad-hoc positioning system. In *Proc. of GlobeCom*, 2001.

[17] D. Niculescu and B. Nath. Ad hoc positioning system (aps) using aoa. In *Proc. of IEEE INFOCOM 2003*, 2003.

[18] G. Pongor. OMNeT: objective modular network testbed. In *MASCOTS'93 Proceedings of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems*, pages 323–326, San Diego, CA, USA, 1993. The Society for Computer Simulation, International.

[19] V. Ramadurai and M. Sichitiu. Localization in wireless sensor networks: A probabilistic approach. In *Proc. of Int. Conf. on Wireless Networks (ICWN)*, 2003.

[20] C. Savarese, K. Langendoen, and J. Rabaey. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proc. of USENIX technical annual conference*, 2002.

[21] A. Savvides, C. Han, and M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proc. of ACM Mobicom'01*, 2001.

[22] A. Savvides, H. Park, and M. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proc. of First ACM Int. Workshop on Wireless Sensor Networks and Application (WSNA)*, 2002.

[23] T.He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proc. of ACM International Conference on Mobile Computing and Networking (Mobicom)*, 2003.

[24] R. Tinos, L. Navarro-Serment, and C. Paredis. Fault tolerant localization for teams of distributed robots. In *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, 2001.

Can an inter-disciplinary research community on location privacy be successful?

Panel

Yugel Saygin
(Panel Moderator)
Sabanci University, Turkey
ysaygin@sabanciuniv.edu

Elisa Bertino
Purdue University, USA
bertino@cs.purdue.edu

Michael Gertz
University of Heidelberg, Germany
michael.gertz@informatik.uni-heidelberg.de

Mohamed Mokbel
University of Minnesota, USA
mokbel@cs.umn.edu

Maria Luisa Damiani
University of Milan, Italy
damiani@dico.unimi.it

1. PANEL OVERVIEW

The newly starting MODAP project (www.modap.org), funded by EU FP7 Future and Emerging Technologies Programme with nearly one million euro funding for three years, aims to coordinate and boost the research activities in the intersection of mobility, data mining, and privacy. The key challenge is to gather an interdisciplinary community of people including, lawyers, psychologists, computer scientists, geographers, and end-users. This panel discusses opportunities, challenges and risks.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining, Spatial databases and GIS; K.4.1 [**Public Policy Issues**]: [Privacy]

General Terms

Management, Security, Legal Aspects

Keywords

Location privacy, Mobility, Data mining

2. MOTIVATION

Capturing the mobility behavior of individuals for on-line or historical data analysis with GPS enabled devices and other positioning systems is an area of increasing interest for both researchers and industry. A typical example from industry is car insurance policy pricing. Recently car insurance companies have started to issue policies with respect to the driving behavior which is captured through a GPS device installed under a special agreement. Traffic authorities are also interested in historical mobility data collected over longer periods for better planning of traffic and emergency scenarios. However, the fact that mobility data is mostly about people associates a high risk with this technology, since it can be used to infer where individuals have been, at what times, how often, and with whom. Therefore, privacy issues need to be addressed in order for the opportunities of mobility data mining to be fully harvested.

A recently completed EU project, GeoPKDD (Geographic Privacy-aware Knowledge Discovery and Delivery, www.geopkdd.eu) was funded by the Future and Emerging Technologies programme which supports high risk, high impact, interdisciplinary research projects. GeoPKDD is one of the first large-scale research projects on mobility data mining with the aim of harvesting knowledge from mobility data. The newly starting MODAP project (www.modap.org), funded by EU FP7 Future and Emerging Technologies Programme with nearly one million euro funding for three years, aims to continue the efforts of GeoPKDD by coordinating and boosting the research activities in the intersection of mobility, data mining, and privacy. For that reason, MODAP aims to create an interdisciplinary platform for technical as well as non-technical people who are interested in mobility data mining together with privacy issues. Such efforts are timely and still needed since privacy risks associated with the mobility behavior of people are still unclear, and it is not possible for mobility data mining technology to thrive without sound privacy measures and standards for data collection, and data/knowledge publishing.

The key challenge is to gather an interdisciplinary community of people including, lawyers, psychologists, computer scientists, geographers, and end-users. As an initial effort to notify the public authorities, a privacy observatory was created within the GeoPKDD consortium. A privacy observatory will also be one of the key bodies of the MODAP project and this time it will include people from different disciplines such as law and ethics. In this panel we will discuss if and how such efforts to build interdisciplinary bodies and communities could be successful in resolving the privacy issues in location data in general and mobility data mining

SPRINGL'09, Nov. 3, 2009, Seattle, WA, USA.

in specific.

Promoting the development of an interdisciplinary community raises important questions about "Why" and "How" to proceed. The ultimate goal is to achieve a comprehensive, holistic view of the privacy issues in mobility which makes possible the development of more effective privacy protection solutions and policies, and at the same time contributes to expand personal and social awareness of privacy rights. Yet, the key question is how to achieve this goal. This question has important research and organizational challenges. From the research viewpoint, a key issue is how to draw the attention of researchers from different disciplines. Which are the benefits that researchers with different backgrounds would obtain from collaboration to the MODAP project? How can researchers agree on which are the research problems that need to be addressed in order to have a real impact and shape this field? Another crucial issue is how to create a platform where these people from different backgrounds can communicate and which kind of initiatives are best suited to foster this collaboration. From an organizational perspective, it is important to establish which institutional bodies and government officials to involve and at what stage. Moreover, what is the "size" of the international dimension, Europe, US, both, Asia? Another concern is what could be an appropriate spectrum of disciplines to consider and which the priorities.

Author Index

Andrienko,Gennady……27
Andrienko,Natalia……...27
Aref ,Walid……..42
Bertino,Elisa ……22, 32, 58
Bringel,José……..12
Capolsini,Patrick……..2
Elmongui, Hicham……...42
Fritsch ,Dieter……..53
Gabillon,Alban…….2
Gertz,Michael……...1
Ghinita,Gabriel…….22
Giannotti,Fosca…….27
Hubig,Christoph……53
Jensen,Christian…….22
Kada,Martin……...53
Kirkpatrick,Michael……...22
Lim,Hyo-Sang……58
Luisa Damiani,Maria…….32
Martin,Hervé……..12
Monga ,Mattia……63
Monreale,Anna……...27
Moon ,Yang-Sae…….58
Ouzzani,Mourad……42
Pedreschi,Dino……..27
Peter ,Michael……53
Ruiz Vicente,Carmen……22
Sicari ,Sabrina……63
Siemoneit ,Oliver……..53
Silvestri,Claudio……32