Access Control Models for Geo-Social Computing Systems

Ebrahim Tarameshloo Philip W. L. Fong
Department of Computer Science,
University of Calgary,
Calgary, Alberta, Canada
{ etarames, pwlfong }@ucalgary.ca

ABSTRACT

A Geo-Social Computing System (GSCS) allows users to declare their current locations, and uses these declared locations to make authorization decisions. Recent years have seen the emergence of a new generation of social computing systems that are GSCSs.

This paper proposes a protection model for GSCSs. The protection system tracks the current locations of users and a knowledge base of primitive spatial relations between locations. Access control policies can be formulated by the composition of primitive spatial relations. The model is extended to account for Geo-Social Network Systems (GSNSs), which track both a spatial knowledge base and a social network. A policy language for GSNSs is proposed for specifying policies that combine both social and spatial constraints.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Access Controls

Keywords

Protection Model, Geo-Social Computing Systems, Geo-Social Network Systems, Location-based Protection, Spatial Relation, Policy Language, Hybrid Logic.

1. INTRODUCTION

With the proliferation of the Internet and GPS enabled smartphones, ***Geo-Social Computing Systems (GSCSs)*** have seen widespread adoption. These systems empower mobile users with knowledge of their vicinity, and thus significantly promote social interactions in contexts including transportation, marketing, health, and the general cultivation of personal and professional relationships. For instance, in PulsePoint, a registered user with cardiopulmonary resuscitation (CPR) training will be notified if a cardiac emergency occurs in his or her neighborhood; in Sonar [37] and Banjo [7], users can meet with friends who are nearby; in Foursquare and Yelp, a user can locate not only nearby friends, but also restaurants and stores with good reviews from friends and other users; in Waze [44], a user selects routes based on traffic reports by friends and other users.

What distinguishes social computing systems is not only the fact that users may contribute personal information to the systems, but also the fact that such user-contributed information is used as the basis of authorization decisions. For example, user relationships are used by social network systems for authorizing accesses [24, 23]. Similarly, a distinguishing feature of GSCSs is that a user may declare her current location (through a mechanism known as "check in"), and such location declarations are used as a basis of authorization (e.g., "allow access if nearby"). GSCS policies can be used for protecting user contributed information, such as photos, status updates, etc. As a special case, the location declaration of a user can also be protected by such policies, just like the friend list can be protected by a relationship-based policy. The focus of this work is the formulation and analysis of access control models in which authorization decisions are a function of location claims.

There has been a growing body of literature on spatially aware access control models [19, 5, 33, 42, 27, 10, 11, 16, 34, 2, 6, 21, 1, 30]. Building on these insights, this study of GSCSs aspires to further our understanding of spatially-aware access control in two areas.

Area 1: Composite Policies. In previous works, location-based policies are usually atomic. In this work, we study how *composite* location-based policies can be formulated in terms of primitive spatial relations. Supporting composite policies offers system designers the flexibility of adopting a larger policy vocabulary, and defining high-level spatial concepts out of low-level spatial relations. Consider a PulsePoint-like GSCS that authorizes a helper's involvement by considering not only the length of the route to the incident location, but also whether the route allows the helper to fetch a nearby AED (automated external defibrillator). This policy involves non-trivial composition of spatial relationships that goes beyond mere Boolean combinations. What kind of composition operators are useful for composing location-based policies? What properties must a composite policy observe in order to reflect common spatial concepts?

Area 2: Geo-Social Network Systems. To the best of our knowledge, there has been no prior work that studies location-based policies in the context of social computing, especially in the context of a ***Geo-Social Network System (GSNS)***, in which authorization decisions are based on both location claims as well as user relationships. For example, in a Yelp-like GSNS, a restaurant may authorize

SACMAT'14, June 25–27, 2014, London, Ontario, Canada.

http://dx.doi.org/10.1145/2613087.2613098 .

a steep promotional discount (as electronic coupons) when a potential customer is not only nearby, but also co-located with three or more friends. What policy languages would support the seamless interleaving of both location and relationship requirements? What authorization architecture is desirable?

In this work, a generic protection model for GSCSs is proposed. Our model captures the two essences of GSCSs: (a) the protection system tracks the declared locations of the users; (b) the relationship between declared locations of the resource requester and the resource owner form the basis of authorization decisions. The spatial knowledge of the protection system is explicitly modelled as a spatial structure (i.e., a graph-like structure; cf. [8]). The latter captures the primitive spatial relationships between known locations. Access control policies can then be formulated by the composition of these primitive spatial relations. The result is a highly flexible theoretical framework for exploring the composition of location-based access control policies. Details of this first contribution are discussed in §3.

A second contribution of this work is presented in §4, in which we propose a classification of GSCSs based on the complexity of the spatial knowledge base. Based on this classification, we outline properties that are common among policies in each family, as well as the implementation strategy for each of the families.

As a third contribution, we identify in §5 the algebraic properties common to reasonable GSCS policies, including co-location, proximity, and policies that interoperate with a spatial hierarchy. Such properties can be used for verifying if a composite GSCS policy has been formulated properly.

These three contributions target **Area 1**.

As a fourth contribution, the GSCS protection model has been extended to account for GSNSs. A GSNS tracks both the declared locations and the declared social relationships of the users. We consider a novel kind of GSNS policies, which are formulated in terms of the social network induced by a spatial neighborhood. A policy language for specifying such policies is proposed, and a modular architecture for authorization is articulated. This contribution, which deepens our understanding of **Area 2**, is detailed in §6.

2. RELATED WORKS

Although the literature is relatively silent on the access control models for GSCSs and GSNSs, there is a vast literature on using location information for controlling access. Spatially aware access control mechanisms can be traced back to [19, 5] (which address the information sharing concerns of satellite images in Geographical Information Systems) or [33, 42] (which propose architectures for location-based applications in wireless local networks).

Generic location-based access control models [27, 11, 34, 21, 30] have been proposed as extensions to the Role-based Access Control (RBAC) [35]. Bertino *et al.* proposed GEO-RBAC [11] to enable RBAC to incorporate spatial restrictions. Specifically, they introduced the notion of *spatial roles*, such that every role is paired with a spatial extent (a set of static spatial boundaries). A spatial role can be enabled only when the user's position is contained in the role's spatial extent. Another extension, LRBAC by Ray *et al.* [34], adds spatial extents to both permissions and roles. Specifically, an object's location must be contained in a permission's spatial extent in order for access to be granted. Kirkpatrick *et al.*, in their work on Prox-RBAC [30], proposed *proximity-based* location constraints to extend RBAC. Specifically, the locations of other users in the system are considered when the access request of a user is authorized. They adopt a spatial model that subdivides an indoor space into a set of *protected areas*. In Prox-RBAC, a proximity constraint considers the *presence* or *absence* of other users in a protected area as well as the continuity of constraint enforcement when users change their locations.

There are also spatiotemporal extensions of RBAC that include both temporal and spatial constraints [16, 2, 6, 1]. In STARBAC [2, 1], the spatial constraints consider only the containment of a physical location in a set of points (as a logical location) for role enabling and disabling. LoT-RBAC [16] is another spatiotemporal extension that employs separate spatial containment hierarchy for each physical and logical location in order to incorporate fine-grained spatial constraints into RBAC. Although the authors were aware of the five topological relations between 2D objects [20], only containment is exploited in their authorization model.

Ardagna *et al.* [3] proposed an access control model for location-based services. They identified three types of location-based conditions. The first type are position-based conditions on the locations of users (e.g., containment or proximity). The second type corresponds to movement-based conditions on the mobility of users (e.g., velocity, acceleration or direction). The third type includes interaction-based conditions among multiple users and entities (e.g., number of users within a given area). Their model supports only a fixed set of location predicate families: i.e., *inarea*, *disjoint* and *distance* for position-based conditions, *velocity* for movement-based conditions, and *density* for interaction-based conditions.

Belussi *et al.* [10] proposed a discretionary access control model for geographical maps stored in spatial databases. Spatial objects have geometric and topological properties. Authorization rules against objects can be specified at a very fine-grained level, and positive and negative authorizations can be propagated among spatial objects.

While the approaches above have made significant contributions in the development of location-based access control models, our work is distinctive in that it captures the richness of spatial reasoning by supporting the formulation of complex location-based policies through the composition of primitive spatial relations. We also demonstrate how this GSCS model can be extended to a GNSN model, thereby allowing us to explore the interplay between spatial awareness and social relationships.

We are aware of an orthogonal previous work [31] that has considered location privacy of members in using proximity services in GSNSs. Their proposed cryptographic-based protocol, only considers *proximity* as an atomic spatial relation between two users. In addition, the *friend* relation is only utilized for identifying the list of users that the secret key should be shared with for a given member.

In recent years, a great deal of attention has been focused on the area of access control models for social computing applications [24, 15, 39, 38, 18]. A Relationship-Based Access Control (ReBAC) model and a series of ReBAC policy languages have been proposed in [23, 25, 13]. There are two points of comparison between this work and that of ReBAC. First, while the spatial structure tracked by a GSCS is mathematically akin to the relational structure in ReBAC,

a novelty of this paper lies in studying the common families of GSCSs (§4) and the common properties of GSCS policies (§5). Second, while hybrid logic is used as a policy language in both ReBAC and GSNS, a novelty of this paper (§6) lies in (a) considering policies that are formulated in terms of the social network induced by a spatial neighborhood, (b) proposing a modular architecture in which the location service is separated from the social network service, and (c) devising an enforcement mechanism for the policies in (a) under the architectural constraints of (b).

The notion of policies induced by path patterns in §5 can be seen as a special case of Fong *et al.*'s notion of inducing an FSNS or ReBAC policy using bi-rooted graph patterns [22, 25]. Cheng *et al.* employs regular expressions as a building block for specifying ReBAC policies that are path based [17]. In comparison, our notion is at the semantic level rather than syntactic level, and thus may capture path pattern set that is not regular.

Jin *et al.* [29] presented a comparative analysis of currently implemented access control mechanisms for the user check-in feature of four GSNSs. They enumerated privacy issues in these GSNSs. Their analysis highlighted the necessity of having a more flexible policy language for GSNSs.

3. A PROTECTION MODEL FOR GEO-SOCIAL COMPUTING SYSTEMS

There are two defining characteristics of a Geo-Social Computing System (GSCS). First, a user can declare her current location through a mechanism commonly known as "check-in." Second, the protection system has a prior notion of how locations are related geometrically, and it uses the relationship between the declared locations of the resource owner and the requester as a basis of authorization.[1] That is, if the declared location of the resource owner and that of the resource requester are related in a way mandated by the access control policy (e.g., co-location, close proximity, spatial containment, etc), then access is granted. The formal model presented below captures this paradigm of access control.

3.1 Notations

Sets and Functions.

We write $|S|$ for the cardinality of set S. Note that if S is infinite, then $|S|$ is a cardinal number [26]. We write 2^S for the powerset of S (i.e., the set of all subsets of S).

We write $\mathcal{F}(S,T)$ for the set of all functions $f : S' \to T$ such that $S' \subseteq S$. That is, $\mathcal{F}(S,T)$ is the set of all *partial* functions from S to T. We write $f : S \rightharpoonup T$ when f is such a partial function. We write $dom(f)$ and $ran(f)$ respectively for the domain and range of function f. Given $f : S \rightharpoonup T$, $s \in S$ and $t \in T$, we write $f[s \mapsto t]$ for the function that maps s to t but otherwise behaves just like f. We write id_S for the identity map over domain S, or simply id if S is known from the context.

[1]The protection mechanism of a geo-social computing system may make use of other information, such as the interpersonal relationship between the resource owner and the requester, as a basis of authorization, but this section focuses on the self-declared location information.

Relational Structures.

Fixing a finite set $\mathcal{I}$ of indices, a ***(binary) relational structure*** is a pair $G = \langle V, \{R_i\}_{i\in\mathcal{I}}\rangle$, where V is a set of entities, and $\{R_i\}_{i\in\mathcal{I}}$ is an indexed family of binary relations such that $R_i \subseteq V \times V$. We write $V(G)$ for V and $R_i(G)$ for R_i. When the set V is finite, then the relational structure is also called a ***graph*** or a ***network***. Then $u \in V$ is a ***vertex*** and $(u,v) \in R_i$ is an ***edge*** of type i. That is, a finite relational structure is an edge-labelled directed graph.

Given a finite set $\mathcal{I}$ and a carrier set $\mathcal{V}$, we write $\mathcal{G}(\mathcal{I},\mathcal{V})$ for the set of all relational structures G with $\mathcal{I}$ as the index set and $V(G) \subseteq \mathcal{V}$. We also write $\mathcal{G}_{\mathsf{fin}}(\mathcal{I},\mathcal{V})$ for the set of all graphs (i.e., finite relational structures) in $\mathcal{G}(\mathcal{I},\mathcal{V})$.

Suppose $G, G' \in \mathcal{G}(\mathcal{I},\mathcal{V})$. A bijective function $f : V(G) \to V(G')$ is called an ***isomorphism*** between G and G' iff $(u,v) \in R_i(G) \Leftrightarrow (f(u),f(v)) \in R_i(G')$. In this case, we say that G and G' are ***isomorphic***, and write $G \equiv_f G'$, or simply $G \equiv G'$ if the identification of f is not important.

Suppose $G = \langle V, \{R_i\}_{i\in\mathcal{I}}\rangle$, and $\mathcal{I}' \subseteq \mathcal{I}$. We write $G \downarrow \mathcal{I}'$ for the graph $\langle V, \{R_i\}_{i\in\mathcal{I}'}\rangle$. That is, $G \downarrow \mathcal{I}'$ is the graph obtained from G by discarding the edges with labels in $\mathcal{I}\backslash\mathcal{I}'$.

Suppose $G = \langle V, \{R_i\}_{i\in\mathcal{I}}\rangle$, and $V' \subseteq V$. The ***subgraph of G induced by V'*** (denoted by $G[V']$) is the relational structure G' for which $V(G') = V'$ and $R_i(G') = R_i(G) \cap (V' \times V')$ for every $i \in \mathcal{I}$.

3.2 Spatial Structures

A GSCS tracks a knowledge base of known locations and their primitive spatial relationships. Access control policies are composed from these primitive relationships. We model the spatial knowledge base as a relational structure $G = \langle L, \{R_i\}_{i\in\mathcal{I}}\rangle$, where $\mathcal{I}$ is a finite set of ***spatial relation identifiers***, and L is a set of known locations. We call G a ***spatial structure***, or a ***spatial network*** in case L is finite. The following are examples of spatial structures.

EXAMPLE 1 (CITIES AND NEIGHBOURHOODS). *Let $\mathcal{I} = \{\mathsf{coloc}, \mathsf{in}, \mathsf{next}\}$ be a set of spatial relation identifiers. Consider a spatial structure $G = \langle L, \{R_i\}_{i\in\mathcal{I}}\rangle$. The locations in L represent either cities (coarser grained location labels) or neighbourhoods (finer grained location labels). R_{coloc} is the co-location relation (i.e., the equality relation, indicating same city or same neighborhood). Also, $(l_1,l_2) \in R_{\mathsf{in}}$ iff neighborhood l_1 is in city l_2. Two neighbourhoods are related by R_{next} whenever they are adjacent to one another. L is a finite set, and thus G is a spatial network.*

EXAMPLE 2 (INDOOR FLOOR PLANS). *An indoor floor plan specified by the space model of [30] can be captured by a spatial network $G = \langle L, \{R_i\}_{i\in\mathcal{I}}\rangle$, where $L = L_{pa} \uplus L_{ep}$ and $\mathcal{I} = \{\mathsf{coloc}, \mathsf{links}, \mathsf{encloses}\}$. The set L is partitioned into two sets. L_{pa} is the set of* **protected areas***, which correspond to enclosed spaces such as rooms, floors, etc. L_{ep} is the set of* **entry points***, which corresponds to, say, doors. The colocation relation R_{coloc} is simply equality. Given an entry point l_1 and a protected area l_2, $(l_1,l_2) \in R_{\mathsf{links}}$ whenever l_1 is an entry point of l_2. Every entry point links exactly two protected areas (i.e., a door links two areas). Given two protected areas l_1 and l_2, $(l_1,l_2) \in R_{\mathsf{encloses}}$ whenever l_1 encloses l_2, and there is no l' such that l_1 encloses l' and l' encloses l_2. R_{encloses} defines a forest.*

EXAMPLE 3 (GPS COORDINATES). *Consider the spatial structure $G = \langle L, \{R_i\}_{i\in\mathcal{I}}\rangle$, where L is the set of GPS coordinates of the form (latitude, longitude), and $\mathcal{I} = \{\mathsf{within\text{-}10}\}$.*

$R_{\mathsf{within\text{-}10}}$ *contains pairs of coordinates that are 10 kilometers apart. Note that L is an uncountable set.*

EXAMPLE 4 (SPATIAL OBJECTS AS POINT SETS). *[20] studies the spatial relationships between three particular kinds of spatial objects in 2-D spaces, namely, points, lines and areas. Examples of such objects may include points of attraction, roads and buildings. Using point set topology and modelling spatial objects as point sets, they show that every two such objects must be related in one of five binary relations:* touch, in, cross, overlap, disjoint. *A spatial knowledge base of a finite number of known 2-D spatial objects can therefore be represented by a spatial network $G = \langle L, \{R_i\}_{i\in\mathcal{I}}\rangle$, in which L is the set of known spatial objects, each of which is a point set, and $\mathcal{I} = \{\mathsf{touch}, \mathsf{in}, \mathsf{cross}, \mathsf{overlap}, \mathsf{disjoint}\}$.*

Point-set topology has been applied for characterizing containment relationships between indoor objects [40] and relationships between spatial objects in an urban area [14].

3.3 Protection State

The protection state of a GSCS is the current location declarations of the users. These location claims are user contributed information, and thus change over time during the normal operation of the GSCS. Formally, if U is the set of all active users (see §3.5 for details), G is the spatial structure tracked by the GSCS, and $L = V(G)$ is the set of all legitimate locations, then the protection state is a function $\eta : U \to L$, which we call ***location assignment***.[2]

3.4 Access Control Policies

Recall that the essence of GSCSs is that authorization decisions are based on the relationship between the declared locations of the resource owner and requester. A "pure" GSCS policy depends only on this location relationship, but not on user identities, roles, attributes, or even interpersonal relationships. Such a "pure" GSCS policy specifies a binary relation between the two locations (that of the resource owner and requester) in the context of a spatial structure. We formalize these notions in the following.

Suppose $\mathcal{I}$ is the set of all spatial relation identifiers, $\mathcal{L}$ is the universe of all location labels, and $\mathcal{U}$ is the universe of all user identifiers. A GSCS policy is a function $P : \mathcal{G}(\mathcal{I},\mathcal{L}) \times \mathcal{F}(\mathcal{U},\mathcal{L}) \to 2^{\mathcal{U}\times\mathcal{U}}$, with some additional requirements to be specified below. The policy $P(G,\eta)$ takes two arguments: (i) a spatial structure $G \in \langle\mathcal{I},\mathcal{L}\rangle$, and (ii) a function $\eta \in \mathcal{F}(\mathcal{U},\mathcal{L})$ that assigns location labels to users. On return, $P(G,\eta) \subseteq \mathcal{U}\times\mathcal{U}$ is a binary relation over users. The additional requirements mentioned above are listed in the following. They are mainly for ensuring the policy behaves in a reasonable way.

1. $P(G,\eta) \subseteq dom(\eta) \times dom(\eta)$. (That is, the policy returns a binary relation over the users with location declarations.)
2. If $ran(\eta) \not\subseteq V(G)$, then $P(G,\eta) = \emptyset$. (That is, if the location assignment is not of the right type, then the policy returns an empty binary relation.)

[2]In some implemented GSCSs, an active user can elect not to declare her current location. Such systems can be modelled by introducing a special location nowhere to indicate an empty declaration. The location nowhere is naturally not related to any location (including itself) in terms of primitive spatial relations. The access control policies (§3.4) can be adjusted accordingly to prevent access if either the owner or requester is at nowhere.

Given an owner u and a requester v, authorization is granted by P iff $(u,v) \in P(G,\eta)$. We write $P(G,\eta)(u,v)$ to assert this condition. Lastly, we write $\mathcal{PO}(\mathcal{I},\mathcal{L},\mathcal{U})$ for the set of all GSCS policies satisfying the above requirements.

A GSCS policy P is ***identity independent*** iff for every spatial structure G, location assignment η, and bijective function $f : dom(\eta) \to dom(\eta)$, we have $P(G,\eta)(u,v)$ whenever $P(G,\eta\circ f^{-1})(f(u),f(v))$. (Here, $\eta\circ f^{-1}$ is the usual functional composition of η with f^{-1}, such that $(\eta\circ f^{-1})(u) = \eta(f^{-1}(u))$.) In other words, permuting user names does not alter authorization decisions. The authorization decisions of an identity-independent policy do not depend on user identities and attributes (e.g., roles).

EXAMPLE 5. *Let P be the following GSCS policy: "Allow access if the owner and the requester are co-located, and no other users are currently located at where they are." That is, $P(G,\eta)(u,v)$ iff $\eta(u) = \eta(v)$ and for every $u' \in dom(\eta)$, $\eta(u') = \eta(u)$ implies that either $u' = u$ or $u' = v$. Policy P is identity independent.*

In the example above, authorization depends not only on the locations of the owner and the requester, but also on the current locations of other users. Many GSCS policies are not like that. In particular, a "pure" GSCS policy authorizes by considering only the relationship between the owner and requester locations, but ignoring the locations of other users. To formalize this idea, we begin with the definition of an auxiliary concept. A ***spatial-relational concept*** is a function $P^* : \mathcal{G}(\mathcal{I},\mathcal{L}) \to 2^{\mathcal{L}\times\mathcal{L}}$, such that $P^*(G) \subseteq V(G)\times V(G)$. That is, P^* maps a spatial structure G to a binary relation $P(G)$ defined over the locations in G. For brevity, we write $P^*(G)(u,v)$ whenever $(u,v) \in P^*(G)$.

A GSCS policy P is said to be ***pure*** iff there exists a spatial-relational concept P^* such that $P(G,\eta)(u,v)$ whenever $P^*(G)(\eta(u),\eta(v))$. By definition, a pure GSCS policy is identity independent.

As we shall see, pure policies are prominent in GSCS implementations. In the rest of this section, as well as in §4 and §5, we will focus on pure policies.

CONVENTION 6. *Unless stated otherwise, GSCS policies are assumed to be pure. For economy of expression, we will identify a spatial relational concept with the pure GSCS policy that the former induces. Therefore, we will assume that a GSCS policy has the same function signature as a spatial relational concept (i.e., $\mathcal{G}(\mathcal{I},\mathcal{L}) \to 2^{\mathcal{L}\times\mathcal{L}}$), and we write $\mathcal{PO}(\mathcal{I},\mathcal{L})$ for the universe of GSCS policies.*

A GSCS policy can be defined as compositions of primitive spatial relations. For example, policies can be formulated as boolean combinations of primitive spatial relations: i.e., union ($R_1 \cup R_2$), intersection ($R_1 \cap R_2$) and complement ($\overline{R}$).

EXAMPLE 7 (CITIES AND NEIGHBOURHOODS). *Suppose an access control policy P is to be formulated for the spatial network of Example 1. Specifically, P grants access if the owner and the requester are either co-located, or located in adjacent neighbourhoods.*

$$P(G) = R_{\mathsf{coloc}}(G) \cup R_{\mathsf{next}}(G) \tag{1}$$

A composite policy can also be formulated via inverse (R^{-1}), relational composition ($R_1 \circ R_2$), transitive closure (R^+) or reflexive transitive closure (R^*).

EXAMPLE 8 (INDOOR FLOOR PLANS). *Consider a policy P for the spatial network of Example 2, such that access is granted if the requester is located in either a protected area l accessible from the protected area in which the owner is located, or in a protected area contained in l.*

$$P(G) = (R_{\mathsf{links}}(G))^{-1} \circ R_{\mathsf{links}}(G) \circ (R_{\mathsf{encloses}}(G))^* \quad (2)$$

Two protected areas l_1 and l_2 are accessible from one another when there exists an entry point l such that $(l, l_1) \in R_{\mathsf{links}}(G)$ and $(l, l_2) \in R_{\mathsf{links}}(G)$. In other words, $(l_1, l_2) \in (R_{\mathsf{links}}(G))^{-1} \circ R_{\mathsf{links}}(G)$. A protected area l_2 is contained in protected area l_1 iff $(l_1, l_2) \in (R_{\mathsf{encloses}}(G))^*$.

CONVENTION 9. *For brevity, we specify policies by mentioning the spatial relation identifiers in place of the actual relations. That is, policies (1) and (2) could have been specified in the following shorthands.*

$$P = \mathsf{coloc} \cup \mathsf{next} \qquad P = \mathsf{links}^{-1} \circ \mathsf{links} \circ \mathsf{encloses}^*$$

CONVENTION 10. *A spatial relational concept or the policy it induces can be seen as a family of binary relations, indexed by spatial structures. Consequently, in this paper we will sometimes talk about spatial relational concepts or policies as if they are binary relations. For example, we might say, "P is reflexive." The intended meaning is that the relation $P(G)$ is reflexive for every G.*

A GSCS policy P is ***topology based*** iff $G \equiv_f G'$ implies that $P(G)(u, v) \Leftrightarrow P(G')(f(u), f(v))$. Topology-based policies are those for which authorization decisions are invariant over isomorphism. The policies in Examples 7 and 8 are both topology based. In §4, we will see that topology-based policies are actually exceptions rather than norms.

3.5 Putting It Together

A GSCS is specified in three "layers." A system schema specifies the ontology of the protection system (i.e., the basic entities that exist in the system). Components of a schema are constant in an installation of the GSCS. A configuration of the system specifies the current privacy settings of the GSCS. Components of a configuration are changed only by administrative operations, though configuration transition is not modelled in this work. Fixing a configuration, a protection state records the system components that may be changed as a result of regular social computing activities. Again, state transition is not the focus of this work.

3.5.1 System Schema

A ***system schema*** (or simply a ***schema***) is a triple $\mathcal{M} = \langle \mathcal{I}, \mathcal{L}, \mathcal{U} \rangle$, where:

- $\mathcal{I}$ is a finite set of spatial relation identifiers
- $\mathcal{L}$ is a set of locations
- $\mathcal{U}$ is a countable set of user identifiers

The sets $\mathcal{I}$ and $\mathcal{L}$ specify the type of spatial relational structures the system tracks. The set $\mathcal{U}$ is the universe of user identifiers. As we shall see, not every user identifier is actively used in the protection state.

3.5.2 Privacy Configuration

A system can be configured with different privacy settings over its life cycle. A ***privacy configuration*** (or simply a ***configuration***) is an abstraction of such settings. Intuitively, a configuration specifies (a) the access control policies of user resources and (b) the spatial knowledge base that defines the spatial relationships among known locations. For part (a), we make the simplifying assumption that there is a single policy that controls the access of all resources owned by a given user. Generalization to per-resource policies is a trivial exercise that does not inspire.

Formally, given a schema $\mathcal{M} = \langle \mathcal{I}, \mathcal{L}, \mathcal{U} \rangle$, a configuration is a tuple $\mathcal{N} = \langle U, L, \{R_i\}_{i \in \mathcal{I}}, policy \rangle$, in which:

- $U \subseteq \mathcal{U}$ is a *finite* set of active users,
- $\langle L, \{R_i\}_{i \in \mathcal{I}} \rangle \in \mathcal{G}(\mathcal{I}, \mathcal{L})$ is a spatial relational structure,
- $policy : U \to \mathcal{PO}(\mathcal{I}, \mathcal{L})$ assigns a policy to each active user.

3.5.3 Protection State

Given a configuration $\mathcal{N} = \langle U, L, \{R_i\}_{i \in \mathcal{I}}, policy \rangle$ of a system schema $\mathcal{M} = \langle \mathcal{I}, \mathcal{L}, \mathcal{U} \rangle$, a ***protection state*** (or simply a ***state***) is a function $\eta : U \to L$ that records the current declared locations of active users.

3.5.4 Authorization

An access request made by a requester v against a resource owned by u is granted if the following check succeeds:

$$P(G)(\eta(u), \eta(v))$$

where $P = policy(u)$ and $G = \langle L, \{R_i\}_{i \in \mathcal{I}} \rangle$. That is, the GSCS will (i) look up the current locations of u and v using the location assignment η, (ii) look up the access control policy $P = policy(u)$ of the resource owner, (iii) instantiate P by the spatial knowledge base $G = \langle L, \{R_i\}_{i \in \mathcal{I}} \rangle$ to obtain a binary relation $P(G)$, and (iv) check if the locations $\eta(u)$ and $\eta(v)$ are related according to binary relation $P(G)$.

3.5.5 A Word on Transitions

There are two levels of dynamism in a GSCS. A ***configuration transition*** occurs when the privacy settings of a system is reconfigured by administrative operations. This may involve (a) introduction or removal of active users in U, (b) introduction or removal of known locations in L, (c) changing the spatial relationships between known labels (i.e., mutating $\{R_i\}_{i \in \mathcal{I}}$), or (d) adopting a different policy $policy(u)$ for some user u. Fixing the system configuration, a ***state transition*** occurs when users change their declared locations in η. State and configuration transitions are not the focus of this paper. We leave this topic to future work.

4. COMMON GSCS FAMILIES

From the four examples of §3.2, we discern three typical families of GSCSs. The classification is based on the cardinality of $\mathcal{L}$, the universe of locations, and the cardinality of L, the set of locations tracked by the spatial knowledge base. We point out in the following the common properties of policies in each family, as well as outlining the implementation strategy of each.

4.1 Logical Locations

Definition. In this family of GSCSs, locations are discrete abstract labels of physical locations (e.g., "111 Lake Louise Drive") such as those in Examples 1 and 2. There are *countably* many such labels in $\mathcal{L}$, but the spatial knowledge base tracks only *finitely* many labels in L (i.e., $|\mathcal{L}| = |\mathbb{N}|$ and $|L| < |\mathbb{N}|$).

Examples. Factual examples of this family of GSCSs include Facebook Places, Foursquare, Yelp, Google Latitude,

Path and Full Circle. Users declare their current locations by selecting a logical label (e.g., a place name or an address) from a list provided by the application. If a user has declared his current location (via check-in), then he can explore friends and places that are in his proximity.

Policies. Policies in a GSCS with logical locations are mostly topology based (see Examples 7 and 8), as the spatial relations are logical rather than geometrical.

Implementation. A typical implementation of such a GSCS stores the entire spatial network. That is, on top of the declared location claims of the users, the graph-like spatial knowledge base is actually tracked by the GSCS in order to support the evaluation of policies. There are general-purpose, efficient procedures for evaluating complex ReBAC policies that are composed from primitive interpersonal relations [17, 13]. Such procedures can be readily adapted for evaluating composite spatial policies in this family.

4.2 Physical Coordinates

Definition. Locations of this family of GSCSs are physical points, such as GPS coordinates, or coordinates in a Euclidean space (Example 3). For these GSCSs, $|\mathcal{L}| = |L| = |\mathbb{R}^k| = |\mathbb{R}|$. That is, the spatial knowledge base is a model of uncountably infinitely many coordinates[3] (though users are located in only finitely many of them).

Examples. A factual example of this family is Sonar [37], which uses the GPS coordinate gathered from a user's smartphone to determine which Sonar users are in close proximity, and thus shall be made visible to that user.[4]

Policies. Policies in a GSCS with locations as physical coordinates are almost never topology-based. Consider the policy within-10 from Example 3. Projection of the GPS coordinates may not preserve the distance between two points.

Implementation. A typical implementation encodes the spatial structure as a set of procedures. Specifically, for each primitive spatial relation, a procedure is implemented for testing if two given points satisfy the primitive spatial relation. Unfortunately, with this implementation strategy, it is unlikely that there exists efficient procedures for evaluating the composition of such primitive spatial relations. The result is that every composition of primitive spatial relations requires a dedicated implementation. This stands in sharp contrast with the case of logical locations.

4.3 Point Sets

Definition. Locations in this third family are spatial objects that correspond to point sets (Example 4). While there are uncountably many possible point sets, the spatial knowledge base tracks only finitely many of them: i.e., $|\mathcal{L}| = |2^{\mathbb{R}}|$, $|L| < |\mathbb{N}|$.

Examples. Waze [44] can be seen as an example of this family. Users can report locations of accidents, traffic jams, speed traps, as well as road hazards and closures to their communities and friends. The reporting mechanism is akin to check-ins, except that the application maps the physical location of the reporting user to the nearest spatial object (i.e., road) as the incident's location. These reports can be accessed by other users as they enter the areas of reported incidents.

Policies. Policies of this family may or may not be topology based. The topology-based policies rely only on the spatial relationships between objects to arrive at an authorization decision. Those that are not may rely on the geometric properties internal to the object itself (e.g., shape, dimensions or size) to make authorization decisions.

Implementation. A typical implementation would precompute the primitive spatial relations between the spatial objects in L, and thus the spatial network is stored as a graph, as in the case of the logical-location family. In this case, there are also general-purpose procedures for evaluating composite policies.

5. VERIFICATION OF GSCS POLICIES

This section discusses the common properties expected of reasonable GSCS policies, particularly those that are formulated in terms of the spatial concepts of proximity, co-location and spatial containment. The goal of this discussion is to provide algebraic tools for assisting a policy engineer in debugging a GSCS policy, by verifying if the draft policy satisfies the aforementioned properties. Detection of violation means that the policy formulation is buggy.

For GSCSs with physical coordinates (§4.2) and point sets (§4.3) as location labels, spatial properties are relatively well understood. For example, axiomatization of the concept of nearness via point-set topology can be found in [43]. The essence of proximity, co-location, and containment are not as well understood in GSCSs with logical location labels and spatial networks. As we shall demonstrate (Example 12), formulation of policies to capture such spatial concepts can be error-prone. Our discussion in this section will therefore focus on GSCSs with a finite universe of location labels.

5.1 Path Patterns

Inspired by [36], we define the notion of path patterns. Given a set $\mathcal{I}$ of relation identifiers, we write $\widetilde{\mathcal{I}}$ to be the set $\left\{\overrightarrow{i} \mid i \in \mathcal{I}\right\} \cup \left\{\overleftarrow{i} \mid i \in \mathcal{I}\right\}$. Here, $\overrightarrow{i}$ is a ***forward edge pattern***, and $\overleftarrow{i}$ is a ***backward edge pattern***. A ***path pattern*** is a finite string of edge patterns. That is, the set $\left(\widetilde{\mathcal{I}}\right)^*$ is the set of all path patterns based on spatial identifiers in $\mathcal{I}$. In particular, the ***empty path pattern*** is denoted by ϵ.

Given a relational structure $G \in \mathcal{G}_{\mathsf{fin}}(\mathcal{I}, \mathcal{V})$, a path p in G is a finite sequence $u_0 u_1 \dots u_n$, such that $n \geq 0$ and $u_i \in V(G)$ for every $0 \leq i \leq n$. The path p is also called a (u_0, u_n)-***path***. The ***length*** of p is n. A ***degenerate path*** is a path with length 0.

We say that a path p ***matches*** a path pattern π in G if there are edges in G along the vertex sequence of p that match the edge patterns in p. Formally, a degenerate path u matches the empty path pattern ϵ; if $p = u_0 u_1 \dots u_n$, $\pi = \overrightarrow{i} \cdot \pi'$, $(u_0, u_1) \in R_i(G)$, and $u_1 \dots u_n$ matches π', then p matches π; if $p = u_0 u_1 \dots u_n$, $\pi = \overleftarrow{i} \cdot \pi'$, $(u_1, u_0) \in R_i(G)$, and $u_1 \dots u_n$ matches π', then p matches π. We write $p \models_G \pi$ when p matches π in G.

Path patterns can be used for specifying a GSCS policy. The GSCS policy ***induced by*** a path pattern set $\Pi \subseteq \left(\widetilde{\mathcal{I}}\right)^*$ is the policy P for which $(u, v) \in P(G)$ iff there exists a (u, v)-path p in G and a path pattern $\pi \in \Pi$ such that

[3] Actual GPS coordinates have limited resolution. Assuming the cardinality of $\mathcal{L}$ to be $|\mathbb{R}|$ underlines the intractability of implementing the spatial structure as a graph.

[4] In Sonar, one can also check in with logical labels (aka places), but that information is not used for access control.

$p \models_G \pi$. A policy induced by a path pattern set is always topology based.

The notion of a GSCS policy induced by a path pattern captures the intuition that many spatial policies grants access when the owner is "accessible" from the requester via a specific type of paths of spatial relationships. The existence of such a path is a proof of accessibility.

5.2 Proximity

A popular access control policy for GSCSs is the proximity policy, which grants access when the owner and the requester are in "close proximity". While there is no standard definition of proximity, there are certain properties that a reasonable proximity policy shall possess. Firstly, the policy must be reflexive: i.e., two co-located persons are in close proximity. Secondly, the policy must be symmetric: i.e., u is close to v if v is close to u. We say that a policy is a ***formal proximity policy*** if it is both reflexive and symmetric.

EXAMPLE 11. *Policy (1) in Example 7 is a formal proximity policy. So is the following policy for Example 3.*

$$P = \mathsf{within\text{-}10}$$

The following is a formal proximity policy for Example 2.

$$P = \mathsf{links}^{-1} \circ \mathsf{links} \tag{3}$$

That a GSCS policy is a formal proximity policy does not mean that it is intended to capture the notion of proximity. Yet, if a policy engineer intends to formulate a GSCS policy to capture the notion of proximity, then she should make sure that the policy is a formal proximity policy, or else the policy is likely to be flawed. Such an error is usually rare when one is working with GSCSs for which location labels are physical coordinates or point sets. When one is working with a GSCS with logical location labels, these types of errors can be subtle, and checking that a policy that is intended to capture proximity is indeed reflexive and symmetric is a first line of defence against errors.

Intuitively, if the requester's location l_2 is "close" to the owner's location l_1 according to a proximity policy P, then as the requester moves from l_2 towards l_1, the requester shall not lose access. We formalize this intuitive notion in the following definition. A GSCS policy P is a ***material proximity policy*** iff (a) P is a formal proximity policy, and (b) P is induced by a path pattern set Π that is prefix-closed. (A path pattern set Π is prefix-closed iff $\pi \in \Pi$ and $\pi = \pi_1 \cdot \pi_2$ jointly imply $\pi_1 \in \Pi$. That is, if a string belongs to a prefix-closed set, then all the prefixes of the string also belong to the set.) The intuition of the definition is that, if $(l_1, l_2) \in P(G)$, then there is a (l_1, l_2)-path that testifies to the "closeness" of l_1 and l_2. As the requester moves from l_2 towards l_1 along this path, all the intermediate vertices are also "close" to l_1. If a policy can be shown to be a material proximity policy, then its formulation is likely to be correct.

EXAMPLE 12. *Among the three policies in Example 11, the only policy that is* not *a material proximity policy is policy (3), which in turn is based on the indoor floor plan domain (Example 2). The policy is formally a proximity policy, and it is induced by a path pattern set* $\Pi = \left\{\overleftarrow{\mathsf{links}} \cdot \overrightarrow{\mathsf{links}}\right\}$. *Nevertheless,* Π *is not prefix-closed.*

To obtain a material proximity policy, we consider the following revision.

$$P = \mathsf{coloc} \cup \mathsf{links} \cup \mathsf{links}^{-1} \cup (\mathsf{links} \circ \mathsf{links}^{-1})$$

The path pattern set Π *to induce* P *is the prefix-closed set below.*

$$\left\{\epsilon, \overrightarrow{\mathsf{links}}, \overleftarrow{\mathsf{links}}, \overleftarrow{\mathsf{links}} \cdot \overrightarrow{\mathsf{links}}\right\}$$

The path pattern set to induce a material proximity policy has some further properties.

PROPOSITION 13. *Let* P *be a material proximity policy that is induced by the path pattern set* Π.

1. $\epsilon \in \Pi$
2. Π *is closed under* **path pattern reversal**.

A pattern set Π is closed under path pattern reversal iff $\pi \in \Pi$ implies $\pi^R \in \Pi$, where the reversal of path pattern π, written π^R, is defined as follows:

$$\epsilon^R = \epsilon \qquad \left(\overrightarrow{i} \cdot \pi\right)^R = \pi^R \cdot \overleftarrow{i} \qquad \left(\overleftarrow{i} \cdot \pi\right)^R = \pi^R \cdot \overrightarrow{i}$$

PROOF. The two properties follow immediately from the reflexivity and symmetry of P. □

5.3 Co-location

Another popular access control policy adopted by simple GSCSs is co-location. A typical co-location policy grants access when the owner and the requester are situated at the same location: i.e., co-location is simply the equality relation. Sometimes, a GSCS may need to capture a less precise notion of co-location in order to promote information disclosure. As in the case of proximity policies, we define in the following reasonable properties that can be expected from policies that are intended to capture the concept of co-location, in GSCSs with logical location labels.

We say that a policy is a ***formal co-location policy*** iff it represents an equivalence relation (i.e., reflexive, symmetric and transitive).

EXAMPLE 14 (CITIES AND NEIGHBOURHOODS). *We formulate the following policy for the GSCS of Example 1, so that access is granted when the requester is located in the same city as the owner.*

$$P = \mathsf{coloc} \cup \mathsf{in} \cup \mathsf{in}^{-1} \cup (\mathsf{in} \circ \mathsf{in}^{-1}) \tag{4}$$

The above policy is a formal co-location policy. Each city, together with its neighbourhoods, form an equivalence class.

By definition, every formal co-location policy is also a formal proximity policy.

A GSCS policy P is a ***material co-location policy*** iff (a) P is a formal co-location policy, and (b) there exists $\mathcal{I}' \subseteq \mathcal{I}$ such that P is induced by the path pattern set $\left(\widetilde{\mathcal{I}'}\right)^*$. An alternative statement of (b) in graph-theoretic terms is that there exists $\mathcal{I}' \subseteq \mathcal{I}$ for which the equivalence classes induced by $P(G)$ are exactly the connected components of $G \downarrow \mathcal{I}'$ (ignoring the directionality of edges).

EXAMPLE 15 (CITIES AND NEIGHBOURHOODS). *Policy (4) in Example 14 is a material co-location policy.*

PROPOSITION 16. *Every material co-location policy is also a material proximity policy.*

PROOF. By definition, a material co-location policy is induced by the path pattern set $\Pi = \left(\widetilde{\mathcal{I}'}\right)^*$ for some $\mathcal{I}' \subseteq \mathcal{I}$. Π is prefix-closed. □

The alternative characterization of material co-location policies given below is more convenient for verification.

PROPOSITION 17. *A GSCS policy P is a material co-location policy iff (a) P is a material proximity policy, and (b) P is transitive.*

The above statement implies that the real difference between a material proximity policy and a material co-location policy is transitivity.

PROOF. ($\Rightarrow$) Suppose P is a material co-location policy. Proposition 16 implies condition (a). That P is a formal co-location policy implies condition (b).

($\Leftarrow$) Suppose P satisfies conditions (a) and (b). Since P is a formal proximity policy (by (a)) and transitive (by (b)), P is also a formal co-location policy.

Since P is a material proximity policy, it is induced by a prefix-closed set Π of path patterns. We claim that if either $\overrightarrow{i}$ or $\overleftarrow{i}$ appears in some path pattern $\pi \in \Pi$, then both $\overrightarrow{i}$ and $\overleftarrow{i}$ belong to Π. Without loss of generality, say $\overrightarrow{i}$ appears in $\pi \in \Pi$. So $\pi = \pi' \cdot \overrightarrow{i} \cdot \pi''$ for some path patterns π' and π''. As Π is prefix-closed, $\pi' \cdot \overrightarrow{i} \in \Pi$. By Proposition 13, Π is closed under path pattern reversal, and thus $\overleftarrow{i} \cdot {\pi'}^R \in \Pi$. By prefix closure again, $\overleftarrow{i} \in \Pi$. Applying closure under path pattern reversal again, $\overrightarrow{i} \in \Pi$ also. In summary, if $\mathcal{I}'$ is the set of all spatial relation identifiers that appear as an edge pattern in Π, then $\widetilde{\mathcal{I}'} \subseteq \Pi$. Since P is reflexive, symmetric and transitive, Π is simply $\left(\widetilde{\mathcal{I}'}\right)^*$. $\square$

5.4 Containment

Spatial containment is a common feature in GSCSs. A ***containment relation*** over locations is (a) a partial ordering (i.e., reflexive, anti-symmetric and transitive), and (b) every location has at most one immediate container. In short, containment induces a spatial hierarchy.

In a GSCS for which its spatial network has a containment relation, policies should be formulated to promote the declaration of fine-grained locations: i.e., if l_1 contains l_2, then we prefer users to declare her current location as l_2 rather than l_1. In particular, a reasonable policy shall grant more access to requesters who declare a finer-grained location.

Suppose R is a containment relation. A policy P is a ***R-consistent*** iff the following holds.

$$P(G)(l_1, l_2) \wedge (l_2, l_2') \in R \Rightarrow P(G)(l_1, l_2')$$

If R is a containment relation for a GSCS, then it is expected that all policies used in the GSCS are R-consistent.

EXAMPLE 18 (INDOOR FLOOR PLANS). *Policy (2) in Example 2 is* $\mathsf{encloses}^*$*-consistent.*

6. A GSNS MODEL

A Geo-Social Network System (GSNS) is an extension of a Geo-Social Computing System (GSCS). A GSNS tracks not only the location claims of users, but also their interpersonal relationships. Both pieces of information will be the basis for authorization decisions. In this section, we will examine how the interplay of the spatial and social dimensions influences the design of a protection system. We will also explore the design of a policy language for specifying GSNS policies.

6.1 Access Control Policies

A GSNS tracks two relational structures: (a) a spatial structure $G = \langle L, \{R_i\}_{i \in \mathcal{I}} \rangle$, and (b) a ***social network***. The latter is a *finite* relational structure $H = \langle U, \{R_j\}_{j \in \mathcal{J}} \rangle \in \mathcal{G}_{\mathsf{fin}}(\mathcal{J}, \mathcal{U})$, where the edge labels come from the set $\mathcal{J}$ of ***social relation identifiers***, and the vertices are from the universe $\mathcal{U}$ of users. Each identifier $j \in \mathcal{J}$ denotes a type of interpersonal relation (e.g., friend, parent, $\mathsf{physician}$, etc). In addition to the two relational structures, a GSNS tracks also the declared location of each user in the social network. As before, this is modelled as a function $\eta : U \to L$. In the next subsection, we will articulate how these three components are assigned to the schema, configuration and state of a GSNS. Before that, we will specify what an access control policy of a GSNS looks like.

As usual, an access request consists of a requester v requesting access to a resource owned by an owner u. A GSNS policy is a function P with the following type signature

$$\mathcal{G}(\mathcal{I}, \mathcal{L}) \times \mathcal{G}_{\mathsf{fin}}(\mathcal{J}, \mathcal{U}) \times \mathcal{F}(\mathcal{U}, \mathcal{L}) \to 2^{\mathcal{U} \times \mathcal{U}} \quad (5)$$

That is, P takes three arguments: (i) a spatial structure $G \in \mathcal{G}(\mathcal{I}, \mathcal{L})$, (ii) a social network $H \in \mathcal{G}(\mathcal{J}, \mathcal{U})$, and (iii) a function $\eta \in \mathcal{F}(\mathcal{U}, \mathcal{L})$ that assigns locations to users. On return, $P(G, H, \eta) \subseteq \mathcal{U} \times \mathcal{U}$ is a binary relation over $\mathcal{U}$. We further require the following of a policy.

1. $P(G, H, \eta) \subseteq dom(\eta) \times dom(\eta)$.
2. If $P(G, H, \eta) \neq \emptyset$ then η must have the function type $V(H) \to V(G)$.

Given an owner u and a requester v, authorization is granted iff $(u, v) \in P(G, H, \eta)$. We write $P(G, H, \eta)(u, v)$ to assert this condition. Lastly, we write $\mathcal{PO}(\mathcal{I}, \mathcal{L}, \mathcal{J}, \mathcal{U})$ for the set of all GSNS policies satisfying the above requirements.

Implicit in the above definition is the possibility for a policy to base authorization decisions on not only the declared locations of the owner and the requester, but also the declared locations of other users in the social network. Consider an alternative (hypothetical) definition of policies in which a policy has the following function type instead:

$$\mathcal{G}(\mathcal{I}, \mathcal{L}) \times \mathcal{G}(\mathcal{J}, \mathcal{U}) \to 2^{\mathcal{L} \times \mathcal{L} \times \mathcal{U} \times \mathcal{U}} \quad (6)$$

That is, a policy takes a spatial network and a social network as input, and produces a 4-ary relation that relates an owner location, a requester location, an owner and a requester. In such a definition, only the locations of the owner and the requester inform the authorization decision. The declared locations of other users do not play a role. As the following example illustrates, there are indeed policies that will take into account the current locations of users other than the owner and the requester.

EXAMPLE 19. *Consider a GSNS akin to Facebook Places, with $\mathcal{I} = \{\mathsf{coloc}\}$ and $\mathcal{J} = \{\mathsf{friend}\}$. Consider the following GSNS policy:*

Policy A: Grant access if requester is both co-located with the owner and also a friend-of-friend of the owner.

This policy is simply a conjunction of a co-location policy and a friend-of-friend policy. Consider the following scenario, which is depicted in Figure 1.

Scenario S: *An owner u and a requester v share exactly one common friend w, such that $\eta(u) = \eta(v) \neq \eta(w)$.*

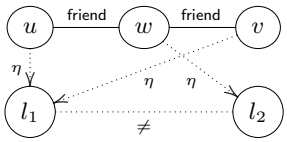

Figure 1: Scenario S

Policy A will grant access in Scenario S. For such kind of policies, the simplistic type signature of (6) would suffice.

In the emerging GSNS, Nextdoor [32], interpersonal relationships are spatially scoped, in the sense that an interpersonal relationship is articulated in the context of a spatial neighborhood (e.g., a city). Generalizing this idea of spatially scoped relationships, consider the GSNS policy below:

Policy B: Grant access if, in the social network of those users who are co-located with the owner, the requester is a friend-of-friend of the owner.

That is, access is only granted if (a) the requester is a friend-of-friend of the owner, and (b) both the requester and one of the common friends between the owner and the requester are co-located with the owner. For the owner u, requester v and common friend w in Scenario S, access will be denied by this policy, because w is not co-located with u. For this kind of policies, the more general type signature of (5) is required.

Policy B above requires that a certain social relationship (friend-of-friend) to hold between the owner and the requester in a spatially constructed social network: i.e., the subgraph of the original social network that is induced by the users co-located with the owner. Such a social network that is induced by a spatial neighborhood of the owner is called a ***spatially scoped social network***. We believe that policies framed in terms of spatially scoped social networks represent a novel protection feature for GSNSs.

6.2 Schemas, Configurations, and States

The specification of the GSNS model closely parallels the three-layer organization of the GSCS model (§3.5). In the following, we highlight the differences between the schemas, configurations and states of GSNSs and those of GSCSs.

A GSNS ***schema*** is a 4-tuple $\mathcal{M} = \langle \mathcal{I}, \mathcal{J}, \mathcal{L}, \mathcal{U} \rangle$. The new component $\mathcal{J}$ is a finite set of social relation identifiers.

A GSNS ***configuration*** $\mathcal{N} = \langle U, L, \{R_i\}_{i \in \mathcal{I}}, policy \rangle$ has the same basic structure as its GSCS counterpart, except that *policy* now has the function type $U \to \mathcal{PO}(\mathcal{I}, \mathcal{L}, \mathcal{J}, \mathcal{U})$. In short, the configuration consists of the spatial structure and user privacy settings.

A GSNS ***state*** is a pair $\langle \eta, \{R_j\}_{j \in \mathcal{J}} \rangle$, where η is the usual location assignment function, and the second component $\{R_j\}_{j \in \mathcal{J}}$ is such that $\langle U, \{R_j\}_{j \in \mathcal{J}} \rangle$ forms a social network. While the spatial structure is a component of the configuration, the social network is a component of the state, which is changeable during the normal operation of the GSNS.

When a requester v attempts to access a resource of owner u, the authorization decision is obtained by evaluating $P(G, H, \eta)(u, v)$, where $P = policy(u)$, $G = \langle L, \{R_i\}_{i \in \mathcal{I}} \rangle$, and $H = \langle U, \{R_j\}_{j \in \mathcal{J}} \rangle$.

6.3 A GSNS Policy Language

We devise a policy language for expressing GSNS policies. The language is intended to be used by system designers and administrators for specification of policies in GSNSs. In this endeavour, we have three design goals in mind. First, previous work [23, 13] has shown that modal logic [12] and hybrid logic [4] form a solid theoretical foundation for the design of a policy language for Relationship-Based Access Control (ReBAC). We therefore take it as a design goal to base the GSNS policy language on the hybrid language of [13], imposing as few perturbations to that language as possible. Second, the GSNS policy language shall support the expression of policies that are based on spatially scoped social networks (Example 19). Third, the language shall support a modular system architecture, in which the location service and the social network system may belong to two separate administrative domains [41].

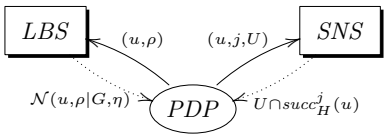

Figure 2: A modular architecture: the Policy Decision Point (PDP) can query a Location-Based Service (LBS) and a Social Network System (SNS).

6.3.1 System Model

We assume a modular design within the Policy Decision Point (PDP). Specifically, we assume the PDP can direct queries to two kinds of services, as shown in Fig. 2: (i) a location-based service (LBS) and (ii) a social network system (SNS). First, the PDP can submit a user u and a spatial relation ρ (e.g., $\mathsf{coloc} \cup \mathsf{in} \cup \mathsf{in}^{-1} \cup (\mathsf{in} \circ \mathsf{in}^{-1})$) to the LBS, and request the LBS to return the set of users v for which the declared locations of u and v are related by the spatial relation ρ (see (7) in §6.3.3). Second, the PDP can submit a user u, a social relation identifier j and a set U of users to the SNS, and request the SNS to return the set of users $v \in U$ for which u and v are related by relation type j (see (8) in §6.3.3). Implicit in this system model is the assumption that location information and social relationships are tracked by two separate subsystems. This arrangement allows a GSNS to selectively "outsource" either the LBS or the SNS (or both) to a different social computing provider [41].

6.3.2 Syntax

A GSNS policy is represented by a formula (ϕ) in a hybrid logic with the following abstract syntax:

$$\rho ::= i \mid -i \mid \overline{\rho} \mid \rho \cup \rho \mid \rho \circ \rho \mid \rho^*$$
$$\phi ::= \top \mid x \mid \neg\phi \mid \phi \wedge \phi \mid \langle j \rangle \phi \mid @_x \phi \mid \downarrow x \,.\, \phi \mid \rho : \phi$$

where $i \in \mathcal{I}$, $j \in \mathcal{J}$ and $x \in X$. The set X is a countably infinite set of variables. We assume that X contains two distinct members, own and req.

Except for the new construct "$\rho : \phi$" and the new syntactic category ρ, the rest of the policy language is essentially the same as the ReBAC policy language of [13]. Each formula ϕ expresses a binary relation between the owner and the requester in a social network. We refer the reader to §6.3.4 or [23, 13] for examples of how these constructs can be used for expressing ReBAC policies, including constraints on depth of friendship or number of common friends.

The newly introduced construct "$\rho : \phi$" is called ***spatial scoping*** (or simply ***scoping***). It is the primary vehicle for

expressing policies based on spatially scoped social networks. Each ***neighborhood expression*** ρ specifies a spatial relation, which is composed from primitive spatial relations (i) using converse ($-$), complement ($\bar{\cdot}$), union ($\cup$), relational composition ($\circ$), or reflexive transitive closure (*). As we shall see below, the semantics of a modal logic is defined via a crawler of the social network. At any point of model checking, the crawler is positioned at a vertex u in the social network. Intuitively, the formula "$\rho : \phi$" means: "*in the subgraph of the social network induced by user u as well as those users who are spatially related to u via the spatial relation ρ, the sub-policy ϕ holds.*"

6.3.3 Semantics

The semantics of our GSNS policy language is specified in three steps: the definition of (1) neighbourhoods, (2) the satisfaction relation, and (3) the authorization relation.

First, we define the meaning of neighborhood expressions. We begin by interpreting each neighborhood expression ρ as a binary relation in a given spatial structure G.

$$[\![i]\!]_G = R_i(G) \qquad [\![-i]\!]_G = R_i(G)^{-1}$$
$$[\![\overline{\rho}]\!]_G = \overline{[\![\rho]\!]_G} \qquad [\![\rho^*]\!]_G = ([\![\rho]\!]_G)^*$$
$$[\![\rho_1 \cup \rho_2]\!]_G = [\![\rho_1]\!]_G \cup [\![\rho_2]\!]_G \quad [\![\rho_1 \circ \rho_2]\!]_G = [\![\rho_1]\!]_G \circ [\![\rho_2]\!]_G$$

Now, the neighborhood of user u induced by the neighborhood expression ρ is the following set.

$$\mathcal{N}(u, \rho \mid G, \eta) = \{v \in dom(\eta) \mid \eta(u) = \eta(v) \vee (\eta(u), \eta(v)) \in [\![\rho]\!]_G\} \quad (7)$$

In short, the neighborhood consists of users whose declared locations are related to the declared location of u through the reflexive closure[5] of $[\![\rho]\!]_G$. Neighbourhoods are essentially obtained by querying the LBS component of the system model (Fig. 2).

Second, we specify the satisfaction relation of hybrid logic: $G, H, \eta, g, U, u \models \phi$. Here, spatial network G, social network H and location assignment η are components from the configuration and state of a GSNS. The ***variable assignment*** $g : X \rightharpoonup V(H)$ interprets variables as users. The set U is a subset of $V(H)$, and $u \in U$ is a user. The definition of the satisfaction relation is specified inductively as follows.

- $G, H, \eta, g, U, u \models \top$ always holds.
- $G, H, \eta, g, U, u \models x$ iff $u = g(x)$[6].
- $G, H, \eta, g, U, u \models \neg\phi$ iff it is not the case that $G, H, \eta, g, U, u \models \phi$.
- $G, H, \eta, g, U, u \models \phi_1 \wedge \phi_2$ iff both $G, H, \eta, g, U, u \models \phi_1$ and $G, H, \eta, g, U, u \models \phi_2$.
- $G, H, \eta, g, U, u \models \langle j \rangle \phi$ iff there exists $u' \in U \cap succ^j_H(u)$ such that $G, H, \eta, g, U, u' \models \phi$. Here, $succ^j_H(u)$ is the set of type-j neighbours of u in the social network H.

$$succ^j_H(u) = \{v \in V(H) \mid (u, v) \in R_j(H)\} \quad (8)$$

  As shown in Fig. 2, the set $U \cap succ^j_H(u)$ is obtained by a query to the SNS component of the system model. Note also that the set $U \cap succ^j_H(u)$ is essentially equivalent to $succ^j_{H[U]}(u)$.

- $G, H, \eta, g, U, u \models @_x \phi$ iff $g(x) \in U$ and $G, H, \eta, g, U, g(x) \models \phi$.
- $G, H, \eta, g, U, u \models \downarrow x \,.\, \phi$ iff $G, H, \eta, g[x \mapsto u], U, u \models \phi$.
- $G, H, \eta, g, U, u \models \rho : \phi$ iff $G, H, \eta, g, U', u \models \phi$ for $U' = U \cap \mathcal{N}(u, \rho \mid G, \eta)$.

Compared to the semantics of the ReBAC policy language of [13], there are three extensions.

1. The rule for scoping ($\rho : \phi$) reduces the current scope U to U', which is obtained by focusing on the spatial neighborhood $\mathcal{N}(u, \rho \mid G, \eta)$ induced by the spatial relation ρ. So the scoping parameter U delimits a smaller and smaller social network as recursion unfolds. Hence, having the scoped social network, the complexity of policy evaluation in our model is at most as hard as policy evaluation in [13].
2. The semantic rules for $\langle j \rangle \phi$ and $@_x \phi$ have been adapted (from their counterparts in [13]) to limit graph traversal within the induced subgraph $H[U]$.
3. Implicit in the semantic rule of x is the requirement that testing succeeds only when $g(x)$ is a vertex in the induced subgraph $H[U]$ (footnote 6).

Third, we specify the authorization relation $G, H, \eta, u, v \Vdash \phi$, which determines if a requester v may access a resource owned by user u, in the context of spatial structure G, social network H, and location assignment η. Specifically, $G, H, \eta, u, v \Vdash \phi$ holds iff $G, H, \eta, g_\star, V(H), u \models \phi$, where

$$g_\star = \{\mathsf{own} \mapsto u, \mathsf{req} \mapsto v\}$$

In short, the global variables own and req denote respectively the owner (u) and the requester (v), and the initial scope is the entire social network ($V(H)$).

6.3.4 Derived Forms and Examples

Standard derived forms can be defined as follows.

$$\rho_1 \cap \rho_2 \stackrel{def}{=} \overline{\overline{\rho_1} \cup \overline{\rho_2}} \qquad \rho^+ \stackrel{def}{=} \rho \circ \rho^*$$
$$\bot \stackrel{def}{=} \neg\top \qquad \phi_1 \vee \phi_2 \stackrel{def}{=} \neg(\neg\phi_1 \wedge \neg\phi_2) \qquad [j]\phi \stackrel{def}{=} \neg\langle j \rangle \neg\phi$$

The following example, taken from [23], reviews how basic modal constructs can be used for expressing interpersonal relationships.

Example 20 (Modal Logic Constructs).

- *Grant access to the owner's spouse:* $\langle \mathsf{spouse} \rangle \mathsf{req}$.
- *Grand parents:* $\langle \mathsf{parent} \rangle \langle \mathsf{parent} \rangle \mathsf{req}$.
- *Parents, aunts and uncles:*

$$\langle \mathsf{parent} \rangle \mathsf{req} \vee \langle \mathsf{parent} \rangle \langle \mathsf{sibling} \rangle \mathsf{req} \vee \langle \mathsf{parent} \rangle \langle \mathsf{sibling} \rangle \langle \mathsf{spouse} \rangle \mathsf{req}$$

- *A sibling who is not married:* $\langle \mathsf{sibling} \rangle (\mathsf{req} \wedge [\mathsf{spouse}] \bot)$.

The following example, adapted from [13], reviews how constructs from hybrid logic can be used for expressing complex graph constraints in the social network.

Example 21. *Grant access if owner and requester share two distinct common friends:*

$$\langle \mathsf{friend} \rangle (\neg \mathsf{own} \wedge \neg \mathsf{req} \wedge \downarrow x \,.\, \langle \mathsf{friend} \rangle (\mathsf{req} \wedge @_{\mathsf{own}} \langle \mathsf{friend} \rangle (\neg \mathsf{own} \wedge \neg \mathsf{req} \wedge \neg x \wedge \langle \mathsf{friend} \rangle \mathsf{req})))$$

The following example illustrates the use of spatial scoping and the @ operator to test simple spatial relation between the requestor and the owner.

[5]Taking the reflexive closure ensures that the location of u is in the neighborhood.

[6]Since it is given that $u \in U$, x is not satisfied if x refers to a vertex outside of $H[U]$ (i.e., $g(x) \notin U$).

Example 22 (Simple Spatial Relation). *The formula below encodes policy (4) from Example 14.*

$$(\mathsf{coloc} \cup \mathsf{in} \cup (-\mathsf{in}) \cup (\mathsf{in} \circ (-\mathsf{in}))) : @_{\mathsf{req}} \top$$

Thanks to the extended semantic rule of $@_{\mathsf{req}}$*, the latter can be used for testing if* req *is in the spatial neighborhood induced by* $\mathsf{coloc} \cup \mathsf{in} \cup (-\mathsf{in}) \cup (\mathsf{in} \circ (-\mathsf{in}))$.

The following example illustrates the checking of a social constellation (i.e., friends of friends) within a spatially scoped social network.

Example 23 (Spatially Scoped Intermediaries). *Consider again the GSNS in Example 19. Policy A can be expressed by the following formula.*

$$(\mathsf{coloc} : @_{\mathsf{req}} \top) \wedge \langle \mathsf{friend} \rangle \langle \mathsf{friend} \rangle \mathsf{req}$$

Policy B, which involves the notion of spatially scoped social networks, can be expressed by the following formula.

$$\mathsf{coloc} : \langle \mathsf{friend} \rangle \langle \mathsf{friend} \rangle \mathsf{req}$$

Due to the extended semantic rule of $\langle \mathsf{friend} \rangle$*, intermediate vertices visited by* $\langle \mathsf{friend} \rangle$ *must fall within the spatial neighborhood induced by* coloc.

The following example illustrates how the nesting of the scoping operator leads to the consideration of a smaller and smaller social network.

Example 24 (Nested Scoping). *Suppose a restaurant owner is to offer promotional discount to a potential customers u who is (a) close to the restaurant, and (b) gathering (i.e., co-located) with three friends. The policy that specifies the availability of promotional discount is encoded by the following formula:*

$$\mathsf{near} : (@_{\mathsf{req}} (\mathsf{coloc} : \mathit{clique}_4))$$

The above formula first focuses on the spatially scoped social network induced by those users near *to the restaurant. It then checks if the requester is within that social network by repositioning the search to* req *(i.e.,* $@_{\mathsf{req}}$*). Now the scope is further reduced to those users who are co-located with the requester (*coloc :*). Lastly it checks if the requester is situated in a clique of size four: i.e., co-located with three friends. The subformula* clique_4 *employs only ReBAC constructs (i.e., no further scoping).*

$$\begin{aligned} &\downarrow x \,.\, \langle \mathsf{friend} \rangle (\neg x \wedge \\ &\quad \downarrow y \,.\, \langle \mathsf{friend} \rangle (\neg x \wedge \neg y \wedge (\langle \mathsf{friend} \rangle x) \wedge \\ &\quad\quad \downarrow z \,.\, \langle \mathsf{friend} \rangle (\neg x \wedge \neg y \wedge \neg z \wedge \\ &\quad\quad\quad (\langle \mathsf{friend} \rangle x) \wedge (\langle \mathsf{friend} \rangle y)))) \end{aligned}$$

7. CONCLUSION AND FUTURE WORK

We proposed an access control model for GSCSs. Complex spatial policies can be composed from primitive spatial relations. We studied the algebraic properties of typical GSCS policies. We extended the model to account for GSNSs. We explored policies that are formulated in terms of spatially scoped social networks, and designed a policy language to capture this notion.

With the advent of Indoor Positioning Systems (IPS) [9] (or High Sensitivity GPS Receivers [46]), future GSCSs/GSNSs may come with much richer spatial models than what is available in existing, GPS-driven systems. We anticipate that future spatial models will involve complex spatial relations induced by indoor or urban settings. A future work is therefore to study the algebraic properties of access control policies in these novel settings.

We assumed that users do not cheat in their location declarations. Several known location verification techniques can be employed in the presence of malicious users [28]. A future direction is to perform location verification by way of social testimony of proximal individuals, and integrate such social testimony with our GSNS authorization scheme.

Another future work is to extend the GSCS/GSNS model to incorporate additional temporal and spatial parameters such as time, direction, orientation and movement [3]. We believe that such extensions would address the need for protection when users and resources are in motion [45].

Acknowledgments

This work is supported in part by an NSERC Discovery Grant and a Canada Research Chair.

8. REFERENCES

[1] S. Aich, S. Mondal, S. Sural, and A. Majumdar. Role Based Access Control with Spatiotemporal Context for Mobile Applications. In *Transactions on Computational Science IV*, volume 5430 of *LNCS*, pages 177–199. Springer Berlin Heidelberg, 2009.

[2] S. Aich, S. Sural, and A. Majumdar. STARBAC: Spatiotemporal Role Based Access Control. In *Proceedings of the OTM'07*, pages 1567–1582. Springer Berlin Heidelberg, Vilamoura, Portugal, 2007.

[3] C. Ardagna, M. Cremonini, S. Capitani di Vimercati, and P. Samarati. Access Control in Location-Based Services. In *Privacy in Location-Based Applications*, LNCS, pages 106–126. Springer Heidelberg, 2009.

[4] C. Areces and B. ten Cate. Hybrid logics. In *Handbook of Modal Logic*, chapter 14. Elsevier, 2007.

[5] A. V. Atluri, V. Atluri, and P. Mazzoleni. A Uniform Indexing Scheme for Geo-spatial Data and Authorizaions. In *Proceedings of the 16th IFIP WG*, pages 207–218. Kluwer Academic Publishers, 2002.

[6] V. Atluri and S. A. Chun. A Geotemporal Role-Based Authorisation System. *IJISC*, 1(1/2):143–168, 2007.

[7] Banjo, Inc. Banjo v3.3, http://www.ban.jo, Apr. 2013.

[8] C. Becker and F. Dürr. On Location Models for Ubiquitous Computing. *Personal and Ubiquitous Computing.*, 9(1):20–31, Jan. 2005.

[9] S. Bell, W. R. Jung, and V. Krishnakumar. WiFi-based Enhanced Positioning Systems: Accuracy Through Mapping, Calibration, and Classification. In *Proceedings of the 2nd ACM SIGSPATIAL*, ISA'10, pages 3–9, San Jose, California, 2010.

[10] A. Belussi, E. Bertino, B. Catania, M. L. Damiani, and A. Nucita. An Authorization Model for Geographical Maps. In *Proceedings of the 12th ACM GIS'04*, pages 82–91, Washington DC, USA, 2004.

[11] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca. GEO-RBAC: A Spatially Aware RBAC. In *Proceedings of the tenth ACM SACMAT*, pages 29–37, Stockholm, Sweden, 2005.

[12] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, New York, 2001.

[13] G. Bruns, P. W. L. Fong, I. Siahaan, and M. Huth. Relationship-based Access Control: Its Expression and Enforcement Through Hybrid Logic. In *Proceedings of the second ACM CODASPY*, pages 117–124, San Antonio, Texas, USA, 2012.

[14] Bucher, B., Falquet, G., Clementini, E., and Sester, M. Towards a Typology of Spatial Relations and Properties for Urban Applications. In *Usage, Usability, and Utility of 3D City Models*. European COST Action TU0801, 2012.

[15] B. Carminati, E. Ferrari, and A. Perego. Enforcing Access Control in Web-based Social Networks. *ACM TISSEC*, 13(1):1–38, Nov. 2009.

[16] S. M. Chandran and J. B. D. Joshi. LoT-RBAC: A Location and Time-Based RBAC Model. In *Proceedings of WISE'05*, pages 361–375. Springer Berlin Heidelberg, 2005.

[17] Y. Cheng, J. Park, and R. Sandhu. A User-to-User Relationship-based Access Control Model for Online Social Networks. In *Proceedings of DBSec'12*, pages 8–24. Springer Berlin Heidelberg, 2012.

[18] Y. Cheng, J. Park, and R. Sandhu. Relationship-Based Access Control for Online Social Networks: Beyond User-to-User Relationships. In *Privacy, Security, Risk and Trust (PASSAT)*, pages 646–655, 2012.

[19] S. A. Chun and V. Atluri. Protecting Privacy from Continuous High-resolution Satellite Surveillance. In *Proceedings of the 14 th IFIP TC11/ WG11.3*, pages 233–244, Deventer, Netherlands, 2001.

[20] E. Clementini, P. D. Felice, and P. v. Oosterom. A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In *Proceedings of SSD'93*, pages 277–295, London, UK, 1993. Springer-Verlag.

[21] I. F. Cruz, R. Gjomemo, B. Lin, and M. Orsini. A Location Aware Role and Attribute Based Access Control System. In *Proceedings of 16th ACM SIGSPATIAL*, pages 84:1–84:2, Irvine, CA, 2008.

[22] P. W. L. Fong. Preventing Sybil Attacks by Privilege Attenuation: A Design Principle for Social Network Systems. In *Proceedings of the S&P'11*, pages 263–278, Oakland, CA, USA, May 2011.

[23] P. W. L. Fong. Relationship-Based Access Control: Protection Model and Policy Language. In *Proceedings of the first ACM CODASPY'11*, pages 191–202, New York, NY, USA, 2011.

[24] P. W. L. Fong, M. Anwar, and Z. Zhao. A Privacy Preservation Model for Facebook-style Social Network Systems. In *Proceedings of the ESORICS'09*, pages 303–320, Berlin, Heidelberg, 2009. Springer-Verlag.

[25] P. W. L. Fong and I. Siahaan. Relationship-based Access Control Policies and Their Policy Languages. In *Proceedings of the 16th ACM SACMAT'11*, pages 51–60, Innsbruck, Austria, 2011.

[26] P. R. Halmos. *Naive Set Theory*. Springer, 1974.

[27] F. Hansen and V. Oleshchuk. Spatial Role-Based Access Control Model for Wireless Networks. In *Proceedings of the 58th IEEE Vehicular Technology Conference*, volume 3, pages 2093–2097, 2003.

[28] W. He, X. Liu, and M. Ren. Location Cheating: A Security Challenge to Location-Based Social Network Services. In *Proceedings of the 31st ICDCS*, pages 740–749, Washington, DC, USA, 2011.

[29] L. Jin, X. Long, J. Joshi, and M. Anwar. Analysis of Access Control Mechanisms for Users' Check-ins in Location-Based Social Network Systems. In *13th IRI'12*, pages 712–717, 2012.

[30] M. S. Kirkpatrick, M. L. Damiani, and E. Bertino. Prox-RBAC: A proximity-based spatially aware RBAC. In *Proceedings of the 19th ACM SIGSPATIAL*, GIS'11, pages 339–348, Chicago, Illinois, 2011.

[31] S. Mascetti, D. Freni, C. Bettini, X. S. Wang, and S. Jajodia. Privacy in Geo-social Networks: Proximity Notification with Untrusted Service Providers and Curious Buddies. *VLDB*, 20(4):541–566, Aug. 2011.

[32] Nextdoor, Inc. http://nextdoor.com, Mar. 2013.

[33] J. Nord, K. Synnes, and P. Parnes. An Architecture for Location Aware Applications. In *Proceedings of HICSS'02*, pages 3805–3810, Washington, DC, USA, 2002. IEEE Computer Society.

[34] I. Ray, M. Kumar, and L. Yu. LRBAC: A Location-aware Role-based Access Control Model. In *Proceedings of the ICISS'06*, pages 147–161, Kolkata, India, 2006. Springer-Verlag.

[35] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-Based Access Control Models. *Computer*, 29(2):38–47, Feb. 1996.

[36] L. Snyder. Theft and conspiracy in the take-grant protection model. *Journal of Computer and System Sciences*, 23(3):333–347, 1981.

[37] Sonar, Inc. Sonar v1.9.3, http://www.sonar.me, 2013.

[38] A. Squicciarini, F. Paci, and S. Sundareswaran. PriMa: An Effective Privacy Protection Mechanism for Social Networks. In *Proceedings of the 5th ACM ASIACCS*, pages 320–323, Beijing, China, 2010.

[39] A. C. Squicciarini, M. Shehab, and J. Wede. Privacy Policies for Shared Content in Social Network Sites. *The VLDB Journal*, 19(6):777–796, Dec. 2010.

[40] E.-P. Stoffel, K. Schoder, and H. J. Ohlbach. Applying Hierarchical Graphs to Pedestrian Indoor Navigation. In *Proceedings of the 16th ACM SIGSPATIAL*, GIS'08, pages 54:1–54:4, Irvine, California, 2008.

[41] E. Tarameshloo, P. W. Fong, and P. Mohassel. On Protection in Federated Social Computing Systems. In *Proceedings of the 4th ACM CODASPY'14*, pages 75–86, San Antonio, Texas, USA, 2014.

[42] U. Varshney. Location Management for Mobile Commerce Applications in Wireless Internet Environment. *ACM Transactions on Internet Technology*, 3(3):236–255, Aug. 2003.

[43] L. S. Vîţă and D. S. Bridges. A Constructive Theory of Point-set Nearness. *Theoretical Computer Science - Topology in Computer Science*, pages 473–489, 2003.

[44] Waze, Inc. Waze v3.6, http://www.waze.com, 2013.

[45] M. Youssef, V. Atluri, and N. R. Adam. Preserving Mobile Customer Privacy: An Access Control System for Moving Objects and Customer Profiles. In *Proceedings of the 6th ACM MDM'05*, pages 67–76, Ayia Napa, Cyprus, 2005.

[46] P. A. Zandbergen and S. J. Barbeau. Positional Accuracy of Assisted GPS Data from High-Sensitivity GPS-enabled Mobile Phones. *Journal of Navigation, Cambridge University Press*, 64:381–399, 7 2011.