Noname manuscript No.
(will be inserted by the editor)

Geo-Social-RBAC: A Location-based Socially Aware Access Control Framework

the date of receipt and acceptance should be inserted later

Abstract The ubiquity of low-cost GPS-enabled mobile devices and the proliferation of online social networks have enabled the collection of rich geo-social information that includes the whereabouts of the users and their social connections. This information can be used to provide a rich set of access control policies that ensure that resources are utilized uniquely and securely. Existing literature focuses on providing access control systems that control the access solely based on either the location of the users or their social connections. In this paper, we argue that a number of real-world applications demand an access control model that effectively captures both the geographic as well as the social dimensions of the users in a given location. We propose, Geo-social-RBAC, a new role based access control model that allows the inclusion of geo-social constraints as part of the access control policy. Our model, besides capturing the locations of a user requesting access and her social connections, includes geo-social cardinality constraints that dictate how many people related by a particular social relation need to be present in the required locations at the time of an access. The model also allows specification of geo-social and location trace constraints that may be used to dictate if an access needs to be granted or denied. Finally, we show that the proposed model is quite expressive and we present a set of conflict resolution schemes to deal with potential undesirable conflicts that may arise in a geo-social context.

1 Introduction

The ubiquity of low-cost GPS-enabled mobile devices and the proliferation of online social networks have enabled the collection of rich geo-social information that includes the whereabouts of the users and their social connections. A number of real-world applications demand an access control model that effectively captures both the geographic as well as the social dimensions of the users in a given location. It is often possible to use this information to help restrict access to a particular set of resources given the location and social context of a user. There are many such scenarios where including location and social relations into access control models may benefit both users and their organizations. For instance, consider a hospital where a doctor can access a patient's record if and only if the doctor is the patient's primary physician and the patient is located in the waiting room outside the doctor's office

Address(es) of author(s) should be given

or when both the patient and the doctor are in the same room. Similarly, we may want to protect the privacy of patients by ensuring that in case a third person enters a room that is not part of the medical personnel and is not the patient's spouse, the health record should be automatically closed to avoid leaking patient's information. Another example of geo-social access controls is restricting confidential information when a stranger enters the same room or automatically opening an office door for a user when some trusted employees located inside the office share a friendship relation with the user.

In addition to geo-locations, location traces also offer interesting potential in the context of geo-social access control. In these cases, the whereabouts of a user and the people she has recently met and the locations she has recently visited would influence how trusted the person is and the access control decision itself. For instance, a trace-based geo-social access control policy may ensure that if a doctor was in a contagious unit, he cannot enter the new born unit unless he goes to a sanitizing facility first. Another example of using geo-social traces to make access decisions will be determining if a user has recently been in company of an untrusted person. This would result in increased suspicion of the user which may indeed require restricting critical privileges to the user. It is also possible in some cases to bootstrap the trust of a user to access a resource based on the people that accompany him and the places where they have been together in the recent past. For instance, in a fast-food restaurant, a user who has just bought something should be allowed to access other areas of the restaurant such as restrooms and in addition, if she also has her kids with her, then she should be allowed to use the kids' play area.

While there are many potential benefits of a geo-social access control model, unfortunately current literature does not provide a solution that allows the specification of such policies which include both geo-social as well as location traces with geo-social cardinality constraints. Most of the existing models support the specification of policies that depend on user location or other contextual factors such as time, type of device used to access the system and the type of connection used to access resources [3,5,16,6,11]. Given that many organizations use role based access control systems (RBAC) [7] to control their resources [12], several existing work have extended this model to include the location context [3,5, 16,11].

In this paper, we propose a fine-grained geo-social access control model, Geo-social-RBAC, that allows the inclusion of geo-social constraints as part of the access control policy.

Concretely, in this paper we make the following *contributions*:

1. To the best of our knowledge, the proposed Geo-social-RBAC model is the first role based access control model that allows the inclusion of geo-social constraints as part of the access control policy.
2. Our model, besides capturing the locations of a user requesting access and her social connections, supports geo-social cardinality constraints that dictate how many people related by a particular social relation need to be present in the required locations at the time of an access. The model also allows specification of fine-grained geo-social and location trace constraints that may be used to dictate if an access needs to be granted or denied based on the historical whereabouts of users.
3. Finally, we show that the proposed model is quite expressive and propose a conflict resolution approach to deal with undesirable conflicts that may arise in a geo-social context.

The remainder of this paper is organized as follows. In Section 2, we discuss the requirements of the system and present an overview of the proposed system. In Section 3, we present the components that we use as part of the system to model the location and social

relations and then introduce the proposed Geo-Social RBAC. In Section 4, we discus possible policy conflicts, and present the proposed conflict resolution approach. In Section 5, we present the related work and we conclude our paper in Section 6.

2 Motivation and Requirements

In this section we motivate the need for the proposed Geo-social RBAC model and present the requirements that guide the design of our geo-social access control framework.

We begin by discussing the types of policies that are unique to the proposed access control model that are not supported by existing systems. As discussed earlier, current access control models do not have the capabilities to support policies that contain geo-social traces and constraints. In this work, we focus on a RBAC [7] based geo-social model as RBAC has become a defacto standard for organizations as it allows the specification of both discretionary and mandatory access control policies and minimizes the administrative effort required to manage access control policies [12]. In RBAC, users are assigned to roles and permissions are assigned to roles. In order to acquire the permissions associated with a role, a user needs to be previously assigned to it and needs to activate it in a session. Conventionally, RBAC does not support location constraints and as a result, several extensions have been proposed to include location constraint [3,5,16,11].

We broadly classify the existing RBAC literature into two categories namely RBAC extensions that support location based decisions [3,5,16,11] such as Geo-RBAC [3] and LoT-RBAC [5] and models that extend RBAC with proximity constraints that include other user's proximity as part of the access control policies such as Prox-RBAC [10] and its extensions [9].

We present a comparison of the access control models in Table 1 based on the following types of policy constraints:

1. *Pure location constraints:* these constraints only take the location of the user into account. For example, to access a confidential file, an engineer may need to be in a specific office room.
2. *Geo-social constraints:* these constraints consider both the location and the social dimensions of the users in the policies. We further classify this type of constraints as follows. *(i) Geo-social graph-based constraint:* these constraints are based on the social graph structure. For instance, to enter into a room a person needs to be in company of at least two friends that work there and are present. *(ii) Geo-social tag-based constraint:* these constraints capture the type of relationships between the users in the social graphs in addition to the location and social constraints. For example, a child can only access a pay-to-view movie if he is in presence of his parent or a nanny. Tag-based policy captures both the knowledge of the social graph and information related to the type of social connection, e.g., nanny.
3. *Trace-based constraints:* These constraints are based on user's trajectory and whether the user has been in contact with a particular set of individuals. We distinguish between two types of constraints. *(i) Location trace-based constraints:* these constraints capture the past location traces of a user as part of the access control policies. For instance, consider a silicon chip manufacture company where even a minimum amount of dust may ruin an entire production batch. If an operator has been in known dusty rooms of the factory, he cannot enter the sterile chip production room unless he has previously passed through the cleaning room. This is a location trace policy as the previous whereabouts of

Policy	RBAC extended with location [3,5,16,11]	RBAC extended with proximity [10,9]	*Our Approach* Geo-Social-RBAC
Pure location constraints	Yes	Yes	Yes
Geo-social graph-based constraints	No	Yes	Yes
Geo-social tag-based constraints	No	No	Yes
Location-trace-based constraints	No	No	Yes
Geo-social-trace-based constraints	No	No	Yes

Table 1 Comparison of types of policies supported by RBAC based systems.

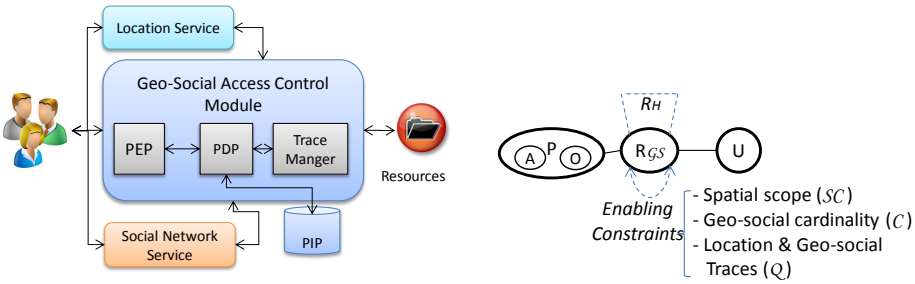
the user determine whether he would be able to obtain the requested access. *(ii) Geo-social trace-based constraints:* these constraints capture both the location history as well as the social dimensions of the users. For example, in a company, if a visitor has entered into the rooms used for induction of new employees accompanied by an administrator, he can also access the welcome package files and the internal directory web pages. In this constraint the location traces of a person are used to determine if an access should be granted. Another scenario where this type of constraint is relevant is in an airport where the whereabouts and the types of people airport personnel visits may influence how trusted they are to access certain resources.

As shown in Table 1, we find that existing models do not support many constraints supported by the proposed Geo-Social-RBAC. With this in mind, we present the following requirements that the geo-social RBAC model addresses. First, the proposed access control framework should allow backward compatibility with RBAC based systems and should effectively support pure location, geo-social and trace-based constraints. The model should allow policies for different spatial granularity, for example, it should be also possible to specify if someone needs to be in a point in the space, at a door, on a room or in a floor of a building, in a city, among others.

2.1 Overview of the Proposed Geo-social RBAC Framework

In Geo-social-RBAC, the context of users is defined by the following information: the position of the user and his previous whereabouts, the proximity of the user to other users and the user's social relations with these individuals. The system consists of *users*, *geo-social roles*, *permissions* and *trace-based* and *geo-social-cardinality constraints*. In our model, users are assigned to geo-social roles and geo-social roles are assigned permissions. To acquire permissions of a geo-social role, users need to be assigned to it and activate it in a session. Geo-social roles can only be activated by a user when his contextual constraints allow it. Hence, a user can only activate a geo-social role when the current location, his previous whereabouts, his proximity to other users and their social relations satisfy the associated activation constraints.

Our framework is depicted in Figure 1. The framework consists of a *Location Service*, one or more *Social Network Services* and a *Geo-Social Access Control Module*. The *Location Service* is trusted to provide the location of the users in the system while the *Social Network Service* is in charge of maintaining and providing the social relations of the users and other information related to the social graph. The *Geo-Social Access Control Module* is responsible for performing access control decisions and consists of the a Policy Enforcement Point (PEP), a Policy Decision Point (PDP), a Policy Information Point (PIP) and a Trace Manager.



(a) Overview of the system components

(b) Graphical representation of the Geo-Social-RBAC

Fig. 1 Framework overview

When a user issues a request to access a resource, the PEP intercepts the request and passes it to the PDP which then makes the decision to grant or deny access to the system based on the policy stored in the PIP and the contextual information of the user. The latter is obtained from the Location and Social Network Services. Based on this information, the Trace Manager evaluates geo-social and location traces constraints.

3 Geo-Social-RBAC

In this section we present the details of the proposed Geo-Social-RBAC. A summary of the notations used in our model is presented in Table 3.

3.1 Social Relations

Without loss of generality, we consider a single social graph that captures the various social relationships among the users. Here, we note that we could also use multiple social graphs services to obtain relevant social information.

Definition 1 Let $\mathcal{G} = \langle V, E \rangle$ be a directed and asymmetric *Social Graph*, where V is a set of vertices and E a set of edges that represent users and their relationships, respectively. We also assume that there is a set of *tags* W used to annotate social relations. For each $e_{(i,j)} \in E$ there is a set that contains one or more tags $W_{(i,j)} \subseteq W$ that denote the type of relation between users i and j. Tags in W are ordered according to a lattice L_W.

A tag represents a specific type of social relation between two users such as a manager-employee relationship, child-nanny relationship, teacher-student relationship and so on. This asymmetry between relations is necessary to ensure that some policies of interest can be specified. For example, suppose $W_{(i,j)} = \{nanny, school\_mate\}$ which shows that user i is the nanny and school mate of user j, while $W_{(j,i)} = \{school\_mate\}$. This allows us to later specify policies of the type "a child cannot access a web page if he is not in presence of his parent or a nanny". Additionally, tags in W may be organized in a lattice L_W that represents a partial order over different relations. For instance, a lattice may show that tags *teacher* and *parent* are greater than tag *student* while *teacher* and *parent* do not have any clear ordered relation, as it is the case when a child request to wash a movie.

Function	*Meaning*
$getSocialRelation : V \times V \rightarrow 2^W$	Returns the tags of a given social relation, e.g., *GetSocialRelation*($v_i \in V, v_j \in V$)= $W_{(i,j)}$.
$getSocialDistance : V \times V \rightarrow \{\mathbb{N} \cup \infty\}$	Returns the minimum number of edges between the specified vertices, e.g., for a direct social relation returns 1, for a friend-of-friend relation returns 2 and for two unconnected nodes ∞.
$superior : V \times V \rightarrow \{t, f\}$	Returns true if the first vertice, v_i, is *superior* to the second vertice, v_j given their tags $W_{(i,j)}$ and lattice L_W.
$commonNeighbors : V \times V \rightarrow \{t, f\}$	Given vertices v_i and v_j returns true if they have neihbors in common, otherwise returns false.
$kClique : 2^V \rightarrow \{t, f\}$	Returns true if the given vertices form a clique, otherwise returns false.

Table 2 Functions to extract relevant information from social graph $\mathcal{G}$.

We use the functions presented in Table 2 to extract relevant information from social graph $\mathcal{G}$. Policies in geo-social-RBAC include relations between a particular user and other users in the social graph. A valid social relation predicate $\mathcal{S}$ is formed by the functions previously listed and allows verification of the existence of a particular(s) social relation(s) or to verify if a social relation has certain properties. Our convention is to denote u as the user and x other collocated individuals, e.g., $superior(u, x)$.

3.2 Geo location and location traces

We make use of the Open GeoSpatial consortium geometric model [1] for modeling the user location and their location traces. In this model, elements in a space called *geometries* are modelled as *points*, *polygons* and *lines*. For instance, a coordinate in the space is a point, a door may be modelled as a line and a building is modelled as a polygon. Additionally, there are several topological relations to relate different geometries in the space; these operations are *overlap*, *touch*, *cross*, *in*, *contains*, *equal*, and *disjoint* and are specified in [1]. Geometries of interest are given names and are called *features*, e.g., a polygon that represents an office may be named office-501. The set of all features of the system is denoted as $\mathcal{F}$.

Definition 2 A *feature* $f \in \mathcal{F}$ is defined as a tuple $\langle type, name \rangle$ where $type \in \{point, line, polygon\}$ represents the geometry type and $name$ represent the name of feature f, respectively.

To model the system, it is necessary to establish a reference space that we denote as $\mathcal{M}$ that provides the limits of the system of interest. We assume that the *Location Service* enables the acquisition of a user u's location as a point in space through a function $location(u)$ that returns a point in $\mathcal{M}$ where user u is or $\perp$ if the user cannot be found.

Let $\mathcal{L}$ be a set of functions to validate the location of users that take as input the location of the user and identify if the location is as expected with respect to a particular place. $\mathcal{L}$ contains operations such as *overlap*, *touch*, *cross*, *in*, *contains*, *equal*, and *disjoint* and may also contain more refined proximity functions as the ones presented in [9]. Note that these functions serve to measure the proximity between a coordinate an a particular location and may be used to establish how far away a user is from others. While $location(u)$ provides coordinates, a function $\ell \in \mathcal{L}$ verifies logical information with respect to a feature f. For example, function ℓ takes the current location of user u, $location(u)$, and a feature and

validates if a user is standing at a particular door. Hence, a tuple $\langle f, \ell \rangle$ defines a spatial scope of interest.

3.2.1 Location traces

Users generate traces as they move around $\mathcal{M}$. We now define different types of traces that may be collected from users.

Definition 3 A *location trace* $\wp l_{(u,t_s,t_e)}$ of user $u \in U$ during a period $[t_s, t_e]$ starting at t_s and ending at t_e is defined as a list $\langle \langle p_1, t_s \rangle, ..., \langle p_i, t_j \rangle, ... \langle p_n, t_e \rangle \rangle$ where each item of the list $\langle p_i, t_j \rangle$ contains location point p_i where user u was at time instance t_j. If at time instance t_j the system has no record of the whereabouts of user u, $p_i = \perp$.

Definition 4 A *geo-social trace* $\wp g_{(u,t_s,t_e)}$ of user $u \in U$ during a period $[t_s, t_e]$ starting at t_s and ending at t_e is a list $\langle \langle p_1, U'_1, t_s \rangle, ..., \langle p_i, U'_i, t_j \rangle, ... \langle p_n, U'_n, t_e \rangle \rangle$ where each item of the list $\langle p_i, U'_i, t_j \rangle$ contains location point p_j where user u was at time instance t_j and $U'_i \subseteq U$ is the set of users that were in proximity as per function $\ell \in \mathcal{L}$ of user u. If at time instance t_j the system has no record of the whereabouts of user u, $p_i = \perp$.

Types of Trace-Based Policies:
We identified two types of trace based policies: *a priori obligation trace* in which a user needs to visit a list of places in order to obtain access and *triggered based policies* in which visiting a location generates a requirement of visiting other locations. An example of *a priori obligation trace policy* is requiring a patient to visit the x-rays room before the first consultation with an orthopaedic physician. An example of a *triggered based policy* is requiring a doctor who has entered into the Contagious Unit to go to the Sanitizing Facility to gain access into the delivery room. Here, the time when a user has visited a place is also relevant; in the example of the doctor entering a delivery unit, we may only care about his whereabouts in the last 24 hours. We use T to denote such period of time, which can be expressed in multiple units such as a week, a day, a number of hours.

Definition 5 A *location trace clause* is a tuple $\langle \alpha, \mathsf{T} \rangle$, where T is a period of time and α is produced by the following grammar[1] where c represents a tuple of type $\langle f \in \mathcal{F}, \ell \in \mathcal{L} \rangle$:
$C ::= C \wedge C \mid C \vee C \mid c$

Definition 6 Similarly, a *geo-social trace clause* is a tuple $\langle \beta, \mathsf{T} \rangle$, where β is produced by the following grammar where g represents a tuple of type $\langle f \in \mathcal{F}, \ell \in \mathcal{L}, s \in \mathcal{S} \rangle$:
$G ::= G \wedge G \mid G \vee G \mid g$

By using the previous definition, it is possible to require users to have come through some places; for instance, requiring a patient to visit the x-rays room or the MRI room before the first consultation with an orthopaedic physician. It is also possible to express this policy in a conjunctive form, to ensure that users go to both places. Similarly, the geo-social trace clause permits the specification of policies where the whereabouts as well as the type of people that the user meets are relevant for making access control decisions. Hence, predicate s is added to indicate the type of social relation.

A location clause $\langle \alpha, \mathsf{T} \rangle$ is fulfilled by user u if his location trace $\wp l_{(u,t_s,t_e)}$, where $\mathsf{T} = [t_s, t_e]$, contains locations that satisfy α. Similarly, a geo-social location clause $\langle \beta, \mathsf{T} \rangle$ is fulfilled if $\wp g_{(u,t_s,t_e)}$ satisfy β.

[1] For simplicity grammars omit the parenthesis to avoid distracting readers from the main issues.

Definition 7 A *trigger location trace clause* is defined as $trigger(\langle f, \ell \rangle, \alpha, \mathsf{T})$ where $\alpha = \perp$ or is produced by the grammar presented in Definition 5 and T is a period of time. Once a user visits $\langle f, \ell \rangle$, he needs to subsequently visit the places specified by α in the recent period T. In case $\alpha = \perp$, regardless what places he visits, he cannot fulfil this clause unless the constraint times out.

Definition 8 A *trigger geo-social trace clause* is defined as $trigger(\langle f, \ell \rangle, \beta, \mathsf{T})$ where β is produced by the grammar in Definition 6 or is equal to $\perp$. Once a user visits $\langle f, \ell \rangle$, he needs to subsequently visit places and people specified by β in the recent period T. In case $\beta = \perp$, regardless of what places and people he visits, he cannot fulfil this clause unless the constraint times out.

Using trigger constraints we can specify policies of the type *if a user enters place x in the recent past, he cannot access object y* by using $\perp$. In the following, we consolidate obligation-based and trigger-based clauses into a single constraint we call a *trace constraint*.

Definition 9 A *trace constraint* $\mathcal{Q}$ is produced by the following grammar where c_1 is a trigger location trace clause, c_2 is a trigger geo-social trace clause, c_3 is a location trace clause and c_4 is a geo-social trace clause (as per Definitions 5, 6, 7 and 8):
$Q ::= Q \wedge Q \mid Q \vee Q \mid T$
$T ::= c_1 \mid c_2 \mid c_3 \mid c_4 \mid \epsilon$

Function *completeTrace* takes as input a trace constraint $\mathcal{Q}$, a user u. If the trace constraint is empty, it returns true. Otherwise, it evaluates each trace clause q in $\mathcal{Q}$ and integrates the results.

3.3 Geo-social Cardinality Constraints

In the proposed model, geo-social cardinality constraints are key to specify whether the locations of a user's social relations should interfere with the access decisions.

Definition 10 A *geo-social cardinality clause* is defined as a tuple $c = \langle f, \ell, n, \mathcal{S} \rangle$ where $f \in \mathcal{F}$ is the feature where at least n social connections that comply with social predicate $\mathcal{S}$ need to be located at according to the proximity function $\ell \in \mathcal{L}$.

Definition 11 A *geo-social cardinality constraint* $\mathcal{C}$ is produced by the following grammar, where c is a geo-social cardinality *clause*:
$C ::= C \wedge C \mid C \vee C \mid Q$
$Q ::= c \mid \epsilon$

We also use function $peopleAt(u, \mathcal{C})$ which takes a user u and a cardinality constraint $\mathcal{C}$ and evaluates if the constraint is satisfied or not. Note that it is possible to have an empty cardinality constraint in which case $peopleAt(u, \mathcal{C})$ returns true.

3.4 Geo-Social-RBAC

With the key building blocks of our model introduced in the previous subsections, we now present the proposed geo-social aware access control model. We first introduce Core-Geo-Social-RBAC and then extend it to include role hierarchy.

Notation	*Meaning*	*Notation*	*Meaning*
$\mathcal{G}$	Social graph.	R	Set of geo-social roles
W	Set of social relation types (tags)	U	Set of uses
L_W	Lattice of social relation types W	A	Set of actions
$\mathcal{S}$	Represents a predicate of social functions	O	Set of objects
$\mathcal{F}$	Features of the system	P	Permissions
$\mathcal{M}$	Reference space	$\mathcal{C}$	Trace constraint
$\mathcal{L}$	Set of location functions	$\mathcal{Q}$	Geo-social cardinality constraint

Table 3 Notation

3.4.1 Core-Geo-Social-RBAC

Core-Geo-Social-RBAC is defined as a tuple $\langle U, R_{\mathcal{GS}}, A, O, P\rangle$. The model consists of a set of geo-social roles $R_{\mathcal{GS}}$, a set of users U, a set of actions A a set of objects O and a set of permissions defined as $P = A \times O$. Users are assigned to geo-social roles and geo-social roles are assigned permissions. We use function $authorized(u \in U)$ to obtain the set of roles that u is authorized for. Figure 1(b) presents a graphical representation of our model.

Definition 12 A *geo-social role* $r \in R_{\mathcal{GS}}$ is defined as a tuple $\langle SC, \mathcal{C}, \mathcal{Q}\rangle$ where

- SC is a set that represents the spatial-scope of a role (places where the role can be activated). The set contains tuples of the form $\langle f \in \mathcal{F}, \ell \in \mathcal{L}\rangle$. When $SC = \perp$ the role does not have a spatial scope is specified.
- $\mathcal{C}$ is a geo-social cardinality constraint as per Definition 11.
- $\mathcal{Q}$ is a trace constraint as per Definition 9.

A geo-social role can be in one of two states *enabled*, or *disable*.

Definition 13 A geo-social role $r = \langle SC, \mathcal{C}, \mathcal{Q}\rangle \in R$ is said to be *enabled* for user u if all the following conditions are fulfilled:

$$r \in authorized(u) \ \wedge \ peopleAt(u, \mathcal{C}) \ \wedge \ completeTrace(\mathcal{Q}, u)$$
$$\wedge \ \exists \langle f, \ell\rangle \in SC \ : \ell(location(u), f) \vee SC = \emptyset$$

Otherwise r is *disabled*.

Henceforth, we refer to *geo-social roles* as *roles*. In the previous definition, a role r is enabled for a user u if u is assigned to r, she is in the required location and the geo-social cardinality and trace constraints are fulfilled. A user u can *activate* role r if it is enabled. When u activates r he can obtain all its privileges. Notice that a geo-social role without any constraint is equivalent to a standard role.

3.4.2 Geo-Social-RBAC with Role Hierarchy

Role hierarchy [13] is a feature used by some RBAC systems in which roles are organized in a partial order. We define a Geo-Social-RBAC system as a tuple $\langle U, R_{\mathcal{GS}}, A, O, P, R_H\rangle$ that contains the same components as core-Geo-social RBAC but also incorporates R_H which symbolizes the geo-social role hierarchy.

Definition 14 (Geo-social Role Hierarchy) Let $r_i, r_j \in R_{\mathcal{GS}}$ be two geo-social roles. r_i is said to be senior of r_j, written as $r_i \geq r_j$. If a user u assigned to r_i can activate r_j as long as r_j is enabled.

Pure location constraint policy: A *researcher* should be in the laboratory (fourth floor) in order to access any general files. Let r_1 be a researcher's geo-social role, with location scope $SC = \langle$*floor4, in*$\rangle$.
Geo-social cardinality constraint(for your eyes only): A *senior-researcher* can access a confidential vaccine compound formula only if he is in the confidential room by himself. Let r_2 be a senior-researcher's geo-social role, with location scope $SC = \langle$*ConfidentialRoom, in*$\rangle$ and a geo-social cardinality constraint $\mathcal{C} = \langle$*ConfidentialRoom*, in, 0, $\epsilon\rangle$.
Geo-social cardinality constraint (tag): An *assistant* in the research lab can only see files with private medical information of subjects if he is in the 4th floor and there are three researchers or senior-researchers (superiors) in the general research unit. Let r_3 be a senior-researcher's geo-social role, with location scope $SC = \langle$*floor4, in*$\rangle$, an a geo-social cardinality constraint $\mathcal{C} = \langle$*GeneralResearchRoom, in, 3, superior(u,x)*$\rangle$.
Location-based trace constraint (trigger): A doctor who was in a contagious unit in the last 24 hours, cannot enter the new born unit unless he goes to a sanitizing facility first. Let r_4 be a geo-social role with $\mathcal{Q} = \langle$*trigger(*$\langle$*ContagiousUnit,in*$\rangle$*,SanitizingFacility,24hours)*$\rangle$.
Trace constraint (obligation): A nurse needs to go to check all patients in their rooms in the last 2 hours before she can sign her electronically the round-sheet. Here, role nurse r_5 is associated with $\mathcal{Q} = (\langle$ *room1,in*$\rangle \wedge \langle$*room2,in*$\rangle \wedge$ ... $\wedge \langle$ *roomn,in*$\rangle$*,2hours*) and with permission sign electronically the round-sheet.

Table 4 Examples of policies that can be expressed in Geo-Social-RBAC.

Note that a user activating r_i does not automatically inherit the permissions of its junior roles. Instead, a user that needs to acquire the permissions of a junior role would need to activate it in a session. This design decision was made to reduce the risk exposure, enforce least privilege and prevent and resolve policy conflicts. A comprehensive discussion that supports this decision is presented in Section 4.

We conclude this section by presenting some examples based on a hospital scenario that has an Emergency Room Unit (E.R.) in the first floor, a Contagious Unit in the second, a New Born Unit in the third floor and a Research Laboratory in the fourth. Based on this scenario in Table 4, we present several examples that demonstrate how our model can be used to express different types of policies.

4 Discussion on Model Expressiveness

In this section, we analyse some key aspects of our model which include its expressiveness, resolution of conflict and discussion on our design.

4.1 Trace Constraints Verification and Enforceability

Trace constraints integrate the allowed or disallowed places that a person may visit alone and who he may meet at certain spots. There are several challenging aspects of this type of policy, mainly related to appropriate clause construction and policy enforcement.

Trace clause construction: We present some properties that allows the detection of inadvertently introduced mistakes in trace clauses and reduce unnecessary verifications during runtime. Recall that all trace clauses (Definitions 5, 6, 7 and 8) contain α and β which define a boolean tree, where $c_i = \langle f_i, \ell_i \rangle$ are nodes and the connections between nodes in the tree are specified by $\wedge$ and $\vee$. We call *trail* a path of nodes joined by $\vee$ in such a tree. Thus, all trails are joined by $\wedge$.

A *well-formed* trace clause complies with the following properties. *(i) Trail minimality:* A trace clause is said to be minimal if each c_i and c_j in a trail do not contain each other. This means that a trail is not composed of features that are contained (in the spatial sense) within each other. For example, it is clear that $\langle\langle floor2, in\rangle \wedge \langle ContagiousUnit, in\rangle\rangle$ is not

a minimal clause, as the second floor contains the Contagious Unit. *(ii) Tree minimality:* A trace clause is said to be tree minimal if the set of nodes that form each trail are not a subset of the nodes of any other trail in the tree. For example, if a tree is formed by $trail_1$ and $trail_2$ which contain $\{c_1, c_2, c_3\}$ and $\{c_1, c_2\}$, the tree is not minimal.
Trace constraints enforceability: First, enforcing trace constraints requires a careful analysis of the paths traveled by the user. For example, consider the case of a physician that enters twice into the Contagious Unit during the specified period window T, the first time she disinfects herself, but the second does not. Such cases require stateful policy enforcement techniques that models the required real-world scenarios captured by the policies while being efficient with respect to performance and scalability. The *Trace Manager* component in our architecture is responsible to ensure that such cases are handled carefully with minimal overhead.

4.2 Tag Lattice Expressiveness

Next, our model incorporates the lattice L_W that has a partial order of tags that annotate relations in a social graph. Related work in the area does not allow such tag-based annotation and relies on RBAC policy to provide information related to the roles that users hold in an organization. We argue that this limits the expressiveness of the model considering that *(i)* RBAC policies usually do not reflect the organization hierarchical structure as they are fine-tuned to enforce least privilege principle. *(ii)* it is also possible for organizations to use core-RBAC systems (that is RBAC without hierarchy) in which case policies that express constraints like *a direct supervisor should be present* are not supported. By introducing L_W, our model supports this type of policies. As the social graph may not always be annotated with the tags that annotate employees' relations, in our model, we achieve the same effect by simply integrating the hierarchical structure of the organization into the access control decision engine to correlate social relations and L_W. We also note that tags in L_W may include any type of attributes that define the relation between users in the system, allowing the definition of very expressive policies.

4.3 Design and Expressiveness of Geo-social Role Hierarchy

In this subsection, we show the reasons for which our geo-social hierarchy is designed to solve possible conflicts that otherwise would occur in policy specifications, without losing expressiveness.

We consider all role hierarchy semantics studied in the literature which were summarized as part of hybrid hierarchy in [13]. Hybrid hierarchy consist of three types of hierarchies: *A-hierarchy*, *I-hierarchy* and *IA-hierarchy*. Consider two geo-social roles r_i, r_j such that $r_i \geq r_j$ and a user u assigned to r_i. In A-hierarchy when u activates r_i does not acquire automatically the permissions of its junior role r_j, he can activate r_j provided that r_j is enabled. In I-hierarchy when u activates r_i gets access to the permissions of both r_i and r_j, but cannot activate r_j. Finally, IA-hierarchy is a combination; activating r_i acquires all permissions from both r_i and r_j and users assigned to r_i may activate r_j. In the following discussion we only analyse I and A hierarchies, as IA is included.

In order to see possible problems that arise when using these types of hierarchical semantics, we first consider the case when r_i can be activated and r_j cannot. For example, r_i should be activated in room-305 when there are at least three people, while r_j can be activated in room-304. In this case, it is clear that a user cannot be simultaneously in both

room-304 and 305. In case of I-hierarchy, when the user is in room-305 he would be able to use all the permissions from r_j. In A-hierarchy, on the other hand, the user would not be able to use these permissions as r_j is not enabled. We argue that from the security perspective that if a role has been assigned a set of constraints its associated permissions should not be available unless the role is enabled. We also argue that this type of semantic is more intuitive for administrators who can be sure that once they have added a constraint to a role, the constraint is always enforced.

We also consider a scenario where r_k is senior of r_m and both of the roles are enabled simultaneously. This would occur if r_k and r_m have the same activation constraints, if they do not have any constraint associated with them or if the constraint of r_k is a sub-case of r_m. For instance, r_k can only be activated when a user is in the third floor alone while r_m can be activated when the user is in room 301, which is in the third floor. In this case, when r_k is activated in I-hierarchy semantics, the permissions of r_m are automatically acquired while in the A-semantics by activating r_k the user does not obtain the permissions of r_n. However, if the user needs to acquire permissions associated with r_k and r_q, she can still activate both roles.

For these reasons, in Definition 14 we use the semantics of A-hierarchies. Additionally, using A-hierarchy semantics reduces the risk exposure significantly as was shown by Baracaldo *et. al* in [2]. We now prove that our model is equally expressive to models that use I-hierarchy.

Theorem 1 *(Hierarchy Expressiveness) The Geo-social hierarchy R_H in Definition 14 can express the same policies as other hierarchy models.*

Proof. Let $r_i, r_j \in R_{GS}$ be related through hierarchical relation $r_i \geq r_j$ and let u be a user assigned to r_i. Additionally, let $P_i \subseteq P$ and $P_j \subseteq P$ denote the set of permissions that are assigned to r_i and r_j, respectively. Since the hierarchy in Definition 14 uses A-semantics, it suffices to prove that for every policy that uses I-hierarchy semantics, there is an equivalent policy in A-semantics that would allow u to acquire the same privileges. Without loss of generality, in the following we assume that r_j does not have any junior roles. Let $P_{au}(r \in R, \diamond)$ denote the permissions that are acquired by user u when activating r and using hierarchy type $\diamond$, where $\diamond = I$ represents the activation of role r when using I-hierarchy semantics, and $\diamond = A$ when using A-semantics. We prove this by cases.

- *Case 1:* r_i and r_j can be activated simultaneously. In this case $P_{au}(r_i, I) = P_i \cup P_{au}(r_i, I)$ and $P_{au}(r_i, A) = P_i$. However, u can easily acquire $P_{au}(r_i, I)$ by activating r_j, which is enabled, hence $P_{au}(r_i, I) = P_{au}(r_i, A), P_{au}(r_j, A)$
- *Case 2:* r_i cannot be activated. In this case $P_{au}(r_i, A) = P_{au}(r_i, A) = \emptyset$.
- *Case 3:* r_i can be activated, but r_j cannot be activated. We divide this case in two sub-cases. *Sub-case 1:* The administrator wants to disallow the access to r_j's permissions when r_j is indeed disabled. In this case, using A-hierarchy achieves the objective, as $P_{au}(r_i, A) = P_i$ while I-hierarchy provides too many permissions. *Sub-case 2:* The administrator wants to provide the permissions of r_j when r_i is activated. In this case, we prove that it is possible to provide $P_{au}(r_i, I) = P_i \cup P_{au}(r_i, I)$ using uniquely A-relations. This can be achieved by creating r_k such that $P_{au}(r_k, A) = P_{au}(r_j, I)$, then u needs to activate r_i and r_k. It is clear that $P_{au}(r_i, A) \cup P_{au}(r_k, A) = P_i \cup P_j$, and hence the claim is proved.

□

Thus, the proposed model is quite expressive and resolves many potential undesirable conflicts that may arise in a geo-social context.

5 Related Work

Several works have extended RBAC to include the context of the user as part of the access control decision [3,5,16,6,11]. In [6] the concept of environmental roles was presented. An environmental role can be activated when a condition takes place. For instance, a role rMonday would be activated when it is Monday. However, this model does not capture the particularities of geo-social constraints or location traces as part of the policies. Other literature [3,5,16,11] focuses in including location as a factor to define access control policies. Geo-RBAC [3] is an access control model in which access decisions depend on the position of the object and the user that is trying to access it. For this purpose, the model includes special roles that are activated when a user is at a particular place. LoT-RBAC [5] presents a similar model that includes temporal constraints in addition to user location. Similarly, in [16] a graph based approach was presented to capture location-and-time based policies. These works differ from ours in that they do not include social relations or geo-location traces as part of their policies.

In the past, some literature [15,8] have propose to include social relations as part of the access control model. TMAC was proposed in [15] to facilitate the establishment of policies that require team cooperation. Users are assigned to teams according to team membership. To get access to a team's resources, user permissions are determined by his or her role and the current activity of the team. Fong present ReRAC [8] where decisions are based on the relationship between the resource owner and the access requester. In contrast to these models, the proposed model considers both geographical and social dimensions of the users for making access decisions. Carminati *et al.* [4] propose an access control system for social computing, user-user and user-resource relationships, and based on which access control policies are formulated. Unlike the proposed geo-social-RBAC model, these works do not include users' geographical context or geo-social traces into the access control model. In [14], a new access control model is presented where access control decisions are made based on the location of the resource owner, the resource requester and possibly other co-located individuals. However, unlike the proposed model, this model considers that individuals own the resources and it is not based on RBAC, making it less suitable for company settings. Also, this model does not consider location trace constraints as captured by our Geo-Social-RBAC model.

Due to the fairly recent popularity of OSNs, only a few works have explored the inclusion of geo-social context as part of access control systems. To the best of our knowledge, the Prox-RBAC model proposed in [10] is the first model to extend the Geo-RBAC model to include proximity of other individuals as part of the policy in indoor environments. However, this model does not allow the specification of geo-social constraints based on social graphs; in Prox-RBAC valid proximity constraints are based on the type of role of other individuals in proximity of the access requester hold. Gupta *et. al* [9] extended this model to provide a proximity based model where policies related to the location, temporal and social proximity in terms of the position between users in a social graph can be specified. Formal definitions to determine the proximity between locations, users, attributes and time, each of which is referred to as a realm are provided. However, their work does not allow the specification of the type of policies that were presented in this paper. More specifically, *(i)* the model presented in [9] does not allow the specification of trace-based constraints that is well captured in our geo-social-RBAC model, *(ii)* unlike our proposed model, the model presented in [9] does not allow the specification of latices to determine partial orders between social relations and *(iii)* finally, we note that the access control model presented in [9] does not include hybrid realm policies. In contrast, our geo-social-RBAC approach allows the specification of both

hybrid realm policies as well as policies based on social tags while also permitting the use of social graph properties to condition access. To the best of our knowledge, the geo-social-RBAC framework presented in this work is the first research effort dedicated to providing a comprehensive role-based access control model that effectively captures both social as well as spatial dimensions of the users considering both geo-cardinality as well location-trace constraints.

6 Conclusions

In this paper, we presented a new access control model that includes geo-social factors of the users as part of the access control decision process. The proposed model allows organizations to specify their policy considering the geographic and social contexts of the access requester users as well as that of the users located near them. We have introduced the concepts of location and geo-location traces, that allow the specification of policies based on the whereabouts of users not only during the access control decision, but during a longer period of time such as their recent past. Our model is compatible with RBAC systems and we believe that it helps mitigate information exfiltration threats and helps better control how users access resources. As part of future work, we are working on devising new techniques and algorithms to efficiently enforce our policy model.

References

1. Opengis simple features specification for sql, technical report ogc 99-049. Technical report, OpenGIS Consortium, 1999.
2. N. Baracaldo and J. Joshi. An adaptive risk management and access control framework to mitigate insider threats. *Computers & Security*, 39:237–254, 2013.
3. E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca. Geo-rbac: a spatially aware rbac. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 29–37. ACM, 2005.
4. B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. A semantic web based framework for social network access control. In *Proc. of the 14th SACMAT*, pages 177–186. ACM, 2009.
5. S. M. Chandran and J. B. Joshi. Lot-rbac: A location and time-based rbac model. In *Web Information Systems Engineering–WISE 2005*, pages 361–375. Springer, 2005.
6. M. J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad, and G. D. Abowd. Securing context-aware applications using environment roles. In *Proc. of the 6th SACMAT*, pages 10–20, 2001. ACM.
7. D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4:224–274, August 2001.
8. P. W. Fong. Relationship-based access control: protection model and policy language. In *Proc. of the first ACM conference on Data and application security and privacy*, pages 191–202. ACM, 2011.
9. A. Gupta, M. S. Kirkpatrick, and E. Bertino. A formal proximity model for rbac systems. *Computers & Security*, 2013.
10. M. S. Kirkpatrick, M. L. Damiani, and E. Bertino. Prox-rbac: a proximity-based spatially aware rbac. In *Proc. of the 19th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, 2011.
11. I. Ray, M. Kumar, and L. Yu. Lrbac: a location-aware role-based access control model. In *Information Systems Security*, pages 147–161. Springer, 2006.
12. Q. M. S. Osborn, R. Sandhu. Configuring role-based access control to enforce mandatory and discretionary access control policies. In *ACM Transaction on Information and System Security*, 2000.
13. R. Sandhu. Role activation hierarchies. In *In Proceedings of 3rd ACM Workshop on Role-Based Access Control*, pages 33–40. ACM, 1998.
14. E. Tarameshloo P. Fong. Access control models for geo-social computing systems. In *SACMAT*, 2014.
15. R. K. Thomas. Team-based access control (tmac): a primitive for applying role-based access controls in collaborative environments. In *Proc. of the 2nd ACM workshop on Role-based access control*, 1997.
16. M. Toahchoodee, I. Ray, and R. M. McConnell. Using graph theory to represent a spatio-temporal role-based access control model. *International Journal of Next-Generation Computing*, 1(2), 2010.