# G-SIR: An Insider Attack Resilient Geo-Social Access Control Framework

Nathalie Baracaldo, Balaji Palanisamy, and James Joshi

**Abstract**—Insider attacks are among the most dangerous and costly attacks to organizations. These attacks are carried out by individuals who are legitimately authorized to access the system. Preventing insider attacks is a daunting task. The recent proliferation of social media and mobile devices offer new opportunities to collect geo-social information that can help in detecting and deterring insider attacks. In particular, such geo-social information allows us to better understand the context and behavior of users. In this paper, we propose a Geo-Social Insider Threat Resilient Access Control Framework (G-SIR) to deter insider threats by including current and historic geo-social information as part of the access control decision process. We include policy constraints to manage the risks of colluding communities, proximity threats, and suspicious users while leveraging the presence of users around the requester to make an access decision. By examining users' geo-social behavior, we can detect those users whose access behavior deviates from the expected patterns; such suspicious behaviors can point to potential insider attackers who may deliberately or inadvertently carry out malicious activities. We use such information to establish how trustworthy a user is before granting access. We evaluate the G-SIR framework through extensive simulations and our results show that the proposed approach is efficient, scalable and effective.

## 1 Introduction

Insider attacks can have significant negative impacts over organizational processes, reputation and revenue [1]. An insider attack is performed by one or more users legitimately authorized to use a system in an organization. Insider attackers use their authorized access to jeopardize the organization by corrupting information, stealing or exfiltrating data or sabotaging operations. Negative impacts of insider attacks include monetary losses, lawsuits, and damage to reputation. In 2014, 28% of the respondents of the US State of Cybercrime Survey [2] reported being victims of insider attacks and 32% reported that insider attacks were more damaging than attacks performed by outsiders. Additionally, 31% of the respondents in that survey reported incidents that could not be attributed with certitude to insiders or outsiders; this indicates a lack of accountability and suggests that there are possibly more incidents caused by insiders.

The increasingly pervasive use of social media and mobile technologies help us gather rich information related to a user's environment. Such information can be used to determine the geo-social context of users to regulate accesses to information assets as well as to identify suspicious or dangerous behavior of users. There are various scenarios in which the geo-social data about individuals can help understand the level of risk associated with granting them access to critical resources or data. For instance, an employee in charge of accessing protected governmental data should not be allowed to fraternize with individuals who belong to groups outside the law, unless their jobs explicitly require those interactions. Geo-social information is also important to determine the required context for granting an access; for example, by identifying where a user is requesting access from and who is in his proximity, it is possible to restrict accesses that are too risky. In some cases, the system may bootstrap the trust of users located around the requester by requiring a particular set of users to be close to the requester at the time an access is requested. In other cases, when a user is requesting access to a confidential resource, such as a sensitive file in the presence of other individuals, the access may need to be denied because of its associated confidentiality disclosure risk.

Despite the availability of data pertaining to the social relationships of users and techniques to analyze it (see [3], [4], [5], [6]), there has been few research efforts that focus on leveraging geo-social information for mitigating insider threats. In this paper, we propose a geo-social insider threat aware methodology and access control (AC) system tailored towards mitigating insider threats. We first present relevant geo-social AC policy constraints which can be used to regulate accesses as well as to flag suspicious insiders. These constraints allow us to identify and prevent accesses that are too risky. We specify the types of geo-social information that can be collected from both inside and outside of an organization and illustrate how they can be incorporated in the AC decision-making process. Rather than specifying the techniques to crawl the data and analyze it from publicly available sources, our goal is to provide a comprehensive model of how to use it after its collection and how to integrate it into an AC decision-making process to mitigate the risk of insider threats.

Some recent work have incorporated geo-social context of users as part of an AC policy [7], [8], [9], however, these existing geo-social AC systems are not designed to take into consideration suspicious user behavior and thus, they are not designed to prevent insider attacks. Evidence shows that technical and psychological precursors can be used to determine when an insider is likely to become an attacker and, therefore, they can help prevent some insider attacks [1]. Hence, by monitoring, analyzing and acting upon sus-

- *All authors are at the University of Pittsburgh.*
  *E-mail addresses: nab62@pitt.edu, bpalan@pitt.edu and jjoshi@pitt.edu*

picious behavior, some insider threats can be mitigated. In various adaptive AC approaches proposed in the literature ( [10], [11], [12], [13], [14], [15]), such information in an AC policy serves to swiftly adapt to negative changes in users' behavior. However, these approaches do not consider geo-social information.

We identify the following key needs related to deterring insider threats using geo-social information. First, to minimize the risk exposure caused by insiders, it is necessary to specify and differentiate between acceptable and unacceptable geo-social behavior. For this purpose, an AC model that allows specification of policies that incorporate this type of information is needed. Using geo-social information, it is possible to determine how trustworthy users are by monitoring the places users have frequented and the interactions between users. Users who consistently meet people and/or visit places they should not, are clearly to be less trusted. Additionally, to further manage the risk exposure, at the time an access request is evaluated, a risk management procedure that considers geo-social aspects should be designed. The risk management process should consider the behavior exhibited by the requesting user, the risk introduced by other users in the vicinity and any indicators of collusion.

The proposed Geo-Social Insider Threat Resilient Access Control Framework (G-SIR) addresses the above needs and is capable of deterring insider attacks by considering users' geo-social context, their behavior and the risks associated with granting access to a set of permissions. The **key contributions** of this paper are summarized as follows:

- To the best of our knowledge, this is the first research effort to analyze geo-social AC systems with the objective of protecting organizations against insider threats. We present threats that are enabled by current geo-social AC systems.
- To mitigate these threats, we propose an AC model that includes a set of geo-social constraints to capture acceptable and unacceptable geo-social behavior. The proposed constraints include geo-social contracts, geo-social obligations, traces and vicinity constraints.
- We propose a risk management framework that incorporates geo-social behavior of the users and adaptably tunes the AC decision to minimize the risk. As part of this process, G-SIR monitors users to identify those who violate geo-social constraints to improve accountability and determine how trustworthy users are. The risk management procedure considers: *i)* how trustworthy a user is with respect to his geo-social behavior, *ii)* the user's current geo-social context, *iii)* the context of relevant social relationships, *iv)* existing indications of collusion among individuals in the vicinity, and *v)* other users in the vicinity who may compromise the security of the information that is being accessed.
- Finally, we evaluate G-SIR through simulations to demonstrate its effectiveness and feasibility.

The remainder of this paper is organized as follows. In Section 2, we present the challenges and requirements. In Section 3, we present some background on geo-social AC systems. In Section 4, we present an overview of the proposed framework. In Section 5, we present in detail our proposed G-SIR model. The risk management procedure is presented in 6. The proposed enforcement algorithm is presented in Section 7. In Section 8, we present system evaluation. In Section 9, we present the related work and we conclude our paper in Section 10.

## 2 Challenges and Requirements

Geo-social information can significantly help deter insider threats. When an organization establishes a geo-social AC system, it creates a unique opportunity to use the information collected by the infrastructure to account for users' behavior and make better AC decisions. These types of controls help prevent some insider attacks. For example, a user who is often at places that he is not supposed to frequent should be flagged as suspicious and actions to restrict his access to highly critical information should be automatically performed. This behavioral information should be considered at the time AC decisions are evaluated.

However, designing a system that uses such information without increasing the risk exposure is a challenging task. Before outlining the concrete challenges and showing where existing techniques fall short, we introduce the relevant actors and components of the proposed system.

### System Actors

A geo-social AC system has a social network graph, where nodes represent users and edges represent relationships among them. These relationships are annotated with labels that represent the types of social relationships. Additionally, a geo-social system has access to the location where users are at any particular time. Users may issue access requests and a policy can be defined to determine if an access request should be granted or denied. Geo-social AC systems also consider where the requester is located and who the users in the vicinity are. This information is very useful because it helps determine when the access request context is not adequate to grant a requested access.

We classify users in the vicinity in three classes: *enablers*, *inhibitors* or *neutral* users according to the way in which they impact the risk exposure associated with granting an access request. *Enablers* are users that may actually bootstrap and/or enhance the trust of an access request by implicitly vouching for the requester due to their social relationship with the requester. *Inhibitors*, on the other hand, are users whose presence increase the risk of granting an access and *neutral* users are those whose presence does not increase or reduce the risk of a request. For example, consider a policy that requires a parent or a nanny to be in the same room with a child requesting an access to a pay-per-view movie. Here, the parent or nanny are enablers and the child is the requester. An example of an inhibitor is a consultant trying to access sensitive information in presence of another consultant working for a competing company.

### Insider Threats

An adaptive geo-social system to deter insider attacks should be able to determine the risks associated with these actors whenever an access request is evaluated. The risk exposure increases with respect to adaptive AC systems because enablers can influence the AC decisions as indicated by the following threats.

**1. Collusion:** The requester and enablers may decide to collude and probe the system to try to access information that they would not ordinarily have access to. Ways to collude to probe the system include changing the current location or trying to modify the social graph to gain more accesses. These types of collusion attacks are new and have not been considered by existing adaptive AC models. Although existing geo-social AC models make use of statically defined and enforced *geo-social cardinality constraints* to reduce the risk of collusion, these constraints are not enough. A geo-social cardinality constraint is a rule that helps establish how many people need to be at a particular location for a user to be able to exercise a privilege [7], [8], [9]. Even if there is evidence that suggests a group of people is colluding, existing geo-social cardinality constraints disregard this information. As a consequence, colluding users may gain access to critical information despite availability of evidence of their malicious efforts.
**2. Social engineering attacks:** Social engineering attackers convince other users to perform an action that they should not perform under normal circumstances. For instance, an enabler may be tricked by a malicious requester through a social engineering attack to move to a location to allow his request to be granted. Similarly, the requester may be tricked to enter into a particular place and access some information.
**3. Proximity threats:** Users in the vicinity create multiple risks based on the groups to which they belong (e.g., conflicting projects, or being part of social communities that are undesirable for a particular access). When a user is in the vicinity of a requester and poses too much risk, she is classified as an inhibitor. A framework for insider attack mitigation needs to be able to specify that whenever there are one or more inhibitors, the access should be denied.
**4. Inadequate policy enforcement:** Although existing geo-social AC systems specify policies that control access to some resources based on the geo-social context of a user, they do not account for negative geo-social behavior. Undesirable behavior may not be prevented by an AC policy for reasons that include high costs of enforcement, inconvenience, and people working around enforcement mechanisms in place, as the following example illustrates. A user may enter a restricted area (e.g., by door piggybacking), where he should not be; however, he does not request any access while in the forbidden place. In this scenario, current geo-social AC systems are blind to the fact that the user entered into a forbidden place. Although it is understood that the user's behavior is inappropriate, no enforced AC policy is impacted by her behavior. Thus, current geo-social AC policies are not enough to detect negative geo-social behavior when it is *not* linked to an access request. As a result, dangerous behavior may not be captured.

Given this inability to enforce desired policies, often users are informed of the geo-social behavior they are expected to fulfill and are blindly trusted to do so. Such desirable behavior can be enforced through *social contracts* [16], which are a tacit or verbal understanding between interested parties about each other's expected behavior. We are interested in social contracts that specify the whereabouts and relationships that are appropriate or inappropriate for the role that users play within an organization.
**5. Privilege misuse threats:** These threats occur when a requester decides to abuse his privileges. Our framework should also mitigate them by using historical behavior.

### Requirements

Towards addressing these insider threats, we now discuss the requirements for the proposed G-SIR framework:

1) Provide policy constructions to classify users in the vicinity as enablers, inhibitors or neutral according to the risk they impose, given an access request.
2) Include policy constraints to capture geo-social behavior relevant to AC decisions. In particular, the system should allow the specification of the following types of policy constraints. *i) geo-social contracts*, which specify places and people that a user cannot visit by virtue of being assigned to a role in an organization, *ii) geo-social obligations*, which are geo-social actions that a user needs to perform after an access has been granted. Geo-social actions include visiting or refraining from visiting a particular place or person, and *iii) trace-based constraints*, which reflect expected paths that users need to complete before being granted an access.
3) Restrict accesses where the requester or any of the enablers are violating any of his social contracts.
4) Monitor and analyze the behavior of users with respect to the fulfillment of geo-social policy constraints. Users violating policy constraints more often than their peers are suspected of disregard of authority and, hence, should be trusted less. Therefore, the estimated probability of the requester being an attacker should include geo-social policy violations.
5) Mitigate the risk posed by colluding users by identifying communities of colluding users and restricting accesses where there is a strong indication that the enablers and the requester are colluding.
6) Ensure that the AC system can adapt to negative changes in behavior of users by restricting critical privileges to users who do not behave properly. The decision to grant or deny an access should consider the risk exposure. G-SIR should minimize the risk exposure caused by the requester, users in the vicinity and potential collusion among enablers and the requester.

## 3 Preliminaries

G-SIR makes use of the notions of *social predicates* and *spatial scopes* introduced in Geo-Social RBAC model [9]. They are defined as follows.
**Social Predicates:** A social graph can be represented as $\mathcal{G} = \langle V, E \rangle$ where $V$ is a set of vertices that represent users and $E$ is a set of edges that represent the existence of a social relation between users. Edges may be also labeled to refine further the types of relationships between users. Let $W$ be a set of social relation labels (e.g., nanny, spouse, etc.) that may be organized in a hierarchy. $W_{(i,j)}$ represents the set of labels of edge $(i, j)$, for example, $W_{(i,j)}$={*nanny*, *aunt*} shows that user $i$ is the nanny and aunt of user $j$.

Additionally, there is a set of social functions to evaluate the social relations between users. Examples of these functions include *areFriends*$(v_i, v_j)$, *haveSocialRelation*$(label, v_i, v_j)$, *socialDistanceLessEqualTo*$(v_i,\ v_j,\ k)$, *isSuperior*$(v_i,\ v_j)$, *haveCommonNeighbor*$(v_i,\ v_j)$, *areInClique*$(v_i,\ v_j)$, *formAClique*($v_i$, $V' \subseteq V$), among others. Functions *belongsToCommunity*($u$, *comm*), and *assignedToRole*($u$, $r$) are useful for our framework and are defined in Table 1. Let $F$ be the set of social functions such as those mentioned above. We define a *Social Predicate* $\mathcal{S}$ as $\mathcal{S} ::= \mathcal{S} \wedge \mathcal{S} | \mathcal{S} \vee \mathcal{S} | f | \neg f$, where $f \in F$ and, for simplicity, parenthesis are omitted. In *social predicates* we use $u_r$ to denote the requesting user and $u?$ to denote a user in the vicinity that is instantiated at the time of evaluation of the policy.

**Spatial Scopes:** A *Spatial Scope*, *SC*, defines a place of interest. It is defined as $SC = \langle h, \ell \rangle$, where $h$ is a feature and $\ell$ is a location function. A feature is a place of interest in space, e.g., room 410, x-y coordinates of a location or hallway. The geometry of these features are defined according to the Open GeoSpatial consortium geometric model [17]. Function $\ell$ evaluates where with respect to *feature* the user needs to be located. For example, $SC = \langle room420, in \rangle$, defines as spatial scope being inside *room420* and $SC = \langle radiusAround(u, 5feet), in \rangle$, defines a circle with a radius 5 feet around the current position of user $u$. Function $\ell$ can also be *overlap*, *touch*, *cross*, *in*, *contains*, *equal*, and *disjoint* [17], and may also be defined using more refined proximity functions as the ones defined in [8].

**Role based access control (RBAC):** RBAC is a widely adopted AC model and has well established advantages [18]. For this reason, G-SIR incorporates its concepts. In RBAC there are permissions, roles, users and sessions. Permissions and users are assigned to roles. 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.

## 4 Overview of the proposed G-SIR

At the core of the proposed G-SIR framework there is an AC policy specification and enforcement mechanism designed to leverage users' geo-social behavior. The AC component captures current and historic geo-social interactions to determine whether an access should be granted or denied. Our framework extends RBAC by allowing the specification and enforcement of geo-social constraints. A role may be subject to the following constraints.

- *Spatial scope:* A role may have a spatial scope that defines a set of locations where it can be activated by users assigned to it.
- *Geo-Social Contracts:* These constraints indicate places that users assigned to the constrained role cannot visit and people they cannot frequently meet.
- *Vicinity constraints:* These constraints impose restrictions on people that may or may not be at a certain distance from the requester at the time of an access. There are two types of vicinity constraints: *inhibiting* and *enabling constraints*. *i) Inhibiting constraints* specify that a requested permission needs to be denied when certain inhibiting users are in the vicinity. These constraints are designed to avoid potential proximity attacks, such as shoulder surfing attacks. For this, a spatial scope where inhibitors cannot be located is defined. *ii) Enabling constraints* are designed to verify the validity of an access request by leveraging the trust on other users in the vicinity of the requester. These constraints specify who and how many people should be in a spatial scope of interest. To enforce them, it is important to ensure that the enablers and the requester are not colluding to prevent insider attacks. We refer to this as *collusion-free* enforcement.

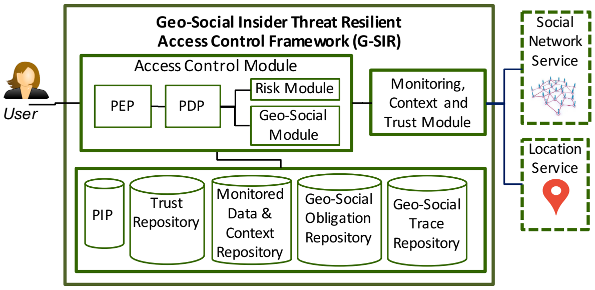

Geo-Social Insider Threat Resilient Access Control Framework (G-SIR)
Access Control Module
User
PEP
PDP
Risk Module
Geo-Social Module
Monitoring, Context and Trust Module
Social Network Service
Location Service
PIP
Trust Repository
Monitored Data & Context Repository
Geo-Social Obligation Repository
Geo-Social Trace Repository

Fig. 1. Overview of the proposed G-SIR framework

- *Geo-social trace based constraints:* These constraints require a user to follow a particular geo-social path before he can be authorized to access a particular resource. They are often useful to ensure that users do not access a resource without proper previous interactions.
- *Geo-social obligations:* These are geo-social actions that users need to fulfill after they have been granted an access.

The proposed constraints are useful in two ways. First, they help capture inappropriate geo-social context and subsequently deny accesses that violate the AC policy. Secondly, monitoring the fulfillment of these constraints provides a way to identify users' whose geo-social behavior is frequently questionable and outside of the expected patterns.

When users violate their geo-social contracts, do not fulfill their obligations or traces, G-SIR flags them as suspicious. Because some of the constraints may be more important than others, their violation has a criticality value. The observations of suspicious geo-social behavior are used to obtain the likelihood of insider attacks and, ultimately, to determine the risk exposure of granting an access.

Figure 1 presents the architecture of G-SIR. All monitoring and likelihood computations described take place in the *Monitoring, Context and Inference Module*. It determines the context of a user, which includes information such as the current device used by the user, type of connection used, etc. The *Access Control Module* is in charge of making the AC decisions. To determine if an access request should be granted, all applicable geo-social constraints are verified. This module also verifies if the risk exposure of granting access to a set of requested permissions is tolerable to allow the access. To manage the risk exposure, at the time of policy specification, the system administrator should perform a utility elicitation process. During this process, described in Section 6, the possible costs of misuse of granting a malicious access, the cost of denying a non-malicious access and gain of allowing a non-malicious access are analyzed. Through this analysis, a threshold that determines the maximum tolerable probability of attack is found. If the probability of attack is too high according to the risk management procedure, the access is denied. Otherwise it is granted.

The steps performed by the *Access Control Module* are as follows. Each access request, $\mathbf{Q}_u$=$\langle u, P' \rangle$, where $u$ denotes the user requesting permission set $P'$, is received by the *Policy Enforcement Point* (PEP). Then, it forwards them to the

*Policy Decision Point* (PDP) which retrieves the policy stored in the *Policy Information Point* (PIP) and evaluates it. An access request is granted by the PDP if the user is assigned to the required roles and all conditions are satisfied; otherwise it is denied.

The implementation of G-SIR requires integrating a location service and a social network service. With respect to OSNs, existing providers such as Facebook and Twitter offer APIs to query the social graph. These APIs are a good way to obtain social information. With respect to location, most organizations are located inside buildings and require the use of indoor location services that may include the use of RFID, Wi-Fi, iBeacons, video cameras, Bluetooth, among others [31]. New methodologies may be developed in the future to systematically deploy reliable and tamper-proof indoor location systems, where users trying to manipulate their reported locations can be identified and flagged as suspicious. This requires orchestrating multiple location technologies to devise a cost-effective deployment solution.

In the next section, we present the proposed AC model.

## 5 G-SIR Access Control Model

The G-SIR access control model consists of sets of roles $R$, users $U$, actions $A$, objects $O$ and sessions $S$. Permissions are defined as $P = A \times O$. Users are assigned to roles, and roles are assigned permissions.

Let $\mathbf{X}$ be a set of contexts dynamically associated with users. The context of user $u$ is denoted as $\mathcal{X}_u$. Let $\mathbf{E}$ be a set of enabling constraints and $\mathbf{I}$ be a set of inhibiting constraints. Additionally, let $\mathbf{GC}$ and $\mathbf{B}$ be the sets of geo-social contracts and geo-social obligations, respectively. Finally, let $\mathbf{W}$ be a set of geo-social traces. All these constraints are formally defined later.

**Definition 1.** *A* role $r \in R$ *in G-SIR access control model is associated with a constraint vector* $CV_r = \langle SC, \mathcal{E}, \mathcal{I}, \mathcal{W}, \mathcal{GC}, \mathcal{B} \rangle$ *where:*

- $SC$ *is the spatial scope where the role can be activated.*
- $\mathcal{E} \subseteq \mathbf{E}$ *and* $\mathcal{I} \subseteq \mathbf{I}$ *represent the constraints enforced over the users in the vicinity, where* $\mathcal{E}$ *defines the required enablers, and* $\mathcal{I}$ *defines inhibitors.*
- $\mathcal{W} \subseteq \mathbf{W}$ *is a set of geo-social trace constraints.*
- $\mathcal{GC} \subseteq \mathbf{GC}$ *is a set of geo-social contracts.*
- $\mathcal{B} \subseteq \mathbf{B}$ *is a set of geo-social obligations.*

To refer to a constraint of a role, we use the dot notation, e.g., $r.SC$ returns the spatial scope of role $r$ and $r.\mathcal{I}$ returns its inhibiting constraint. In section 5.6, we specify how a role can be activated in a session. We make use of the functions presented in Table 1. We also use the dot notation to refer to components of tuples. We now define the above constraints.

### 5.1 Geo-social Contracts

Geo-social contracts are used to establish acceptable and unacceptable geo-social behavior for different roles. Geo-social contracts are assigned to a user when he is assigned to a role. These contracts need to be fulfilled at all times.

**Definition 2.** *A* Geo-Social Contract $gc \in \mathbf{GC}$ *is defined as* $gc = \langle \omega, \varphi \rangle$ *where*

- $\omega = \langle SC, \mathcal{S} \rangle$, *here* $SC$ *represents a spatial scope indicating places that users subject to gc are not allowed to visit, and* $\mathcal{S}$ *is a social predicate that defines undesirable acquaintances. When a component in* $\omega$ *is set to* $\perp$, *(e.g.,* $\omega.SC = \perp$*), it indicates that it is not considered during the enforcement.*
- $0 \leq \varphi \leq 1$ *represents how critical it is for the organization if a user violates the contract. Here,* $\varphi = 1$ *means that it is very critical while* $\varphi = 0$ *means not critical at all.*

If user $u$ is assigned to a role set $R_u \subseteq R$, to be allowed to activate any role in $R_u$, he needs to fulfill all geo-social contracts associated with each role in $R_u$.

**Example 1.** *(a) Consider a user Bob who is assigned to role* secretary; *by being assigned to this role, he cannot access a laboratory where highly reactive chemicals are located because he is not trained to deal with dangerous chemicals. If Bob accesses the lab, there is an inherent risk of mishandling substances that may lead to accidents and loss of lives and intellectual property. For this reason, a violation of this contract will result in a high risk for the organization. This constraint can be expressed as follows:* $gc_1 = \langle\langle\langle chemicalLab, in\rangle, \perp\rangle, 0.9\rangle$. *(b) Consider a consulting firm that may have projects from multiple competing companies, say* $X$ *and* $Y$. *The consulting firm needs to ensure that the projects are completely compartmentalized to be able to offer a quality consulting service. Besides enforcing separation of duty –where no user can be assigned both roles, namely consultant for* $X$, $r_x$, *and consultant for* $Y$, $r_y$ *– it is desirable that people belonging to conflicting projects are not together to avoid leakage of information. Contractors that have multiple clients often require this type of control. These constraints can be expressed as follows:* $gc_2 = \langle\langle\perp, \mathit{assignedToRole}(u?, r_x)\rangle, 0.5\rangle$ *and* $gc_3 = \langle\langle\perp, \mathit{assignedToRole}(u?, r_y)\rangle, 0.5\rangle$. $gc_2$ *is associated with role* $r_y$ *and* $gc_3$ *is associated with role* $r_x$. *As the previous scenarios show, not all contracts are the same in terms of risk exposure. An untrained person entering a lab that has a lot of volatile chemicals poses higher risk compared to the same person entering into a meeting room reserved for a team working in a classified advertisement. Hence,* $gc_1.\varphi > gc_2.\varphi$.

### 5.2 Vicinity Constraints

Inhibiting and enabling constraints are designed to classify users in the vicinity as enablers, inhibitors or neutral.

**Definition 3.** *An* Inhibiting Constraint $ci \in \mathbf{I}$ *is defined as tuple* $\langle X, SC, \mathcal{S}, \alpha \rangle$ *where*

- $X \subseteq \mathbf{X}$ *is a subset of contexts where the inhibiting constraint is applicable,*
- $SC$ *defines the spatial scope where the inhibiting constraint is evaluated,*
- $\mathcal{S}$ *defines the predicate used to classify users in the vicinity as inhibitors and*
- $\alpha$ *is a threshold to determine the minimum level of confidence needed to decide if a user should be made part of the inhibiting group.*

*We say that if there is a set of one or more users* $U_{ci}$ *in location* $ci.SC$, *who fulfill social predicate* $ci.\mathcal{S}$ *with a minimum confidence level of* $\alpha$, *the constraint is not satisfied and the access should be denied to prevent potential information leakage.*

At the policy evaluation time, G-SIR verifies if the requester's context is one of the context specified in $X$. If it is, the inhibiting constraint is evaluated otherwise it is ignored. This helps specify policies where the device the user is utilizing may influence the size of the spatial scope evaluated as illustrated in the following example.

| Function | Meaning |
|---|---|
| $assigned(u \in U)$ | Returns the set of roles that $u$ is assigned to. |
| $P_{au}(r \in R)$ | Returns the set of permissions assigned to $r$. |
| $P_{au}(R' \subseteq R)$ | Returns the set of permissions assigned to all roles in $R'$. |
| *validLocation*$(u, r)$ | Returns true if the current location of $u$ satisfies the spatial scope of $r.SC$. |
| *vicinity*$(SC)$ | Returns a set of users located in the place specified by spatial scope $SC$. |
| *PrCollusion*$(U_c \subseteq U)$ | Returns the probability that users in $U_c$ are colluding. |
| *belongsToCommunity*$(u \in U,$ *comm*$)$ | Given a user $u$ and a community *comm*, returns true if the user is part of *comm*. |
| *assignedToRole*$(u \in U, r \in R)$ | Given a user $u$ and a role $r$, returns true if $u$ is assigned to $r$. |
| *fulfillSocialPredicate*$(u_r, u_c, \mathcal{S})$ | Returns true if users $u_c$ and $u_r$ fulfill social predicate $\mathcal{S}$. |
| *fulfillContracts*$(u \in U)$ | Returns true if user $u \subseteq U$ currently satisfies *all* his contracts. It evaluates the union of all contracts assigned to roles in $assigned(u)$. |
| $inhibitors(u_r \in U, r \in R)$ | Given a requester $u_r$ and a role $r$, evaluates each $ci \in r.\mathcal{I}$ and returns the union of all users classified as inhibitors as per Definition 3. If $inhibitors(u_r, r) = \emptyset$, all $ci \in r.\mathcal{I}$ are satisfied. |
| $enablers(u_r \in U, r \in R)$ | Given a requester $u_r$ and a role $r$, returns a set $U_e \subseteq U$, if it exists, that satisfies all enabling constraints $r.\mathcal{E}$ according to Definition 5. Otherwise it returns $\emptyset$. If $r.\mathcal{E} = \emptyset$, it returns $\emptyset$. |
| $completeTraces(r \in R, u \in U)$ | Returns true if user $u$ has completed traces $r.\mathcal{W}$. |
| $traceContains(w \in \mathcal{W}, node)$ | Returns true if trace $w$ contains $node$ as part of its spatial scope $w.SC$. |
| $disjoint(SC_i, SC_j)$ | Given two spatial scopes $SC_i$ and $SC_j$, returns true if the $SC_i$ is disjoint in $SC_j$. |
| *fulfillO*$(u \in U, r \in R)$ | Returns true if user $u$ satisfies obligations $r.\mathcal{B}$ and *all* his geo-social contracts. |

TABLE 1
Function specifications for G-SIR.

**Example 2.** *Assume* smartphone, laptop *and* presenter *are contexts of interest. Consider a requester $u_r$ who is assigned to role $r_1$ with inhibiting constraints $r_1.\mathcal{I} = \{ci_1, ci_2\}$, where $ci_1 = \langle\{laptop, smartphone\}, \langle radiusAround(u_r, 5feet), in\rangle\rangle$, $belongToCommunity(u?, BadGuys), 0.95\rangle$ and $ci_2 = \langle\{presenter\}, \langle conferenceRoom, in\rangle, belongToCommunity(u?, BadGuys), 0.95\rangle$. When $u_r$ is using a laptop or smartphone, $ci_1$ is evaluated to verify the presence of inhibitors within a 5feet radius. If $u_r$ is using a presenter, $ci_2$ is evaluated to verify that no inhibitors are present in the conference room. In both $ci_1$ and $ci_2$, a vicinity user is classified as inhibitor if he belongs to the community BadGuys with a confidence level of 0.95 or more.*

**Definition 4.** *An* Enabling Constraint $ce \in \mathbf{E}$ *is defined as a tuple $\langle SC, k, \mathcal{S}, \tau_c\rangle$ such that $SC$ is a spatial scope where $k$ users who fulfill social predicate $\mathcal{S}$ with respect to the requester need to be located, and $0 \leq \tau_c \leq 1$, is a threshold that defines the maximum tolerance for colluding users.*

Here, $ce.\tau_c$ is the maximum acceptable probability of collusion and should be specified based on the risk of an access. A larger $ce.\tau_c$ reflects more tolerance to collusive behavior. In fact, if $ce.\tau_c = 1$, the collusion indicators are not considered at all. In contrast, when $ce.\tau_c = 0$ any suspicion of collusion results in invalidating a set of enablers. Threshold $ce.\tau_c$ provides a way to determine when a set of potential enablers cannot be trusted. It is compared with the value obtained by function *PrCollusion*, which depends on the organization's activities and can be found through methodologies such as those presented in [19], [20]. Consider a candidate set of enablers $U_e$, if *PrCollusion*$(U_e)$ $> ce.\tau_c$, the candidate enablers are rendered untrustworthy.

$ce.\mathcal{S}$ may be evaluated based on uncertain information. For example, a social graph may be evaluated to identify if users belong to dangerous communities through algorithms such as those presented in [4], [5], [6]. These algorithms output a set of communities and a confidence level of the result. $ce.\alpha$ determines the minimum confidence level required to classify a user as part of a community. In contrast, when $ce.\mathcal{S}$ is evaluated based on information that is well-established, $ce.\alpha$ can be set to one.

**Example 3.** *Consider role $r_2$ with a set of enabling constraints $r_2.\mathcal{E} = \{ce_1\}$. Enabling constraint $ce_1$ is defined as: $\langle\langle conferenceRoom, in\rangle, 4, areFriends(u?, u_r), 0.8\rangle$. $ce_1$ requires four users who are friends of the requester to be in the conference room and for them to be non-colluding with a probability of 0.8 or more.*

**Definition 5.** *Given a requester $u_r$, an enabling constraint $ce = \langle SC, k, \mathcal{S}, \tau_c\rangle$ is said to be* satisfied *if and only if there exists a set of enablers $U_e$ such that $\forall\ u_e \ \in U_e\ :$*

1) $u_e \in vicinity(ce.SC)$ .
2) $fulfillSocialPredicate(u_r, u_e, ce.\mathcal{S})$
3) *PrCollusion*$(U_e \cup u_r) \leq ce.\tau_c$
4) $fulfillContracts(u_e)$
5) $|U_e| \geq ce.k$

In the previous definition, the risk of including invalid enablers is controlled in two ways. *i)* by verifying that the probability of collusion between the set of enablers is less than the specified confidence threshold and *ii)* by verifying that none of the enablers is violating any of his contracts. This mitigates potential social engineering attacks where an enabler is tricked into going to the required spatial scope $ce.SC$ to satisfy enabling constraint $ce$. It similarly thwarts attacks where the requester and enablers probe the system to see what accesses they can obtain.

**Conflict Resolution:** Because inhibiting and enabling constraints are evaluated dynamically based on who is located in the vicinity at the time of the access request, it is possible that one or more users in the vicinity may be classified as both inhibitor and enabler. We call this a vicinity conflict. It arises when for a given role, $inhibitors(u_r, r) \cap enablers(u_r, r) \neq \emptyset$. For $ce \in r.\mathcal{E}$ and $ci \in r.\mathcal{I}$, recall that $ce.\mathcal{S}$ specifies social relations of the users, whereas $ci.\mathcal{S}$ specifies users in the vicinity suspected of belonging to dangerous or undesirable communities for an access. Hence, a user may be related to another and at the same time be suspected of participating in a non-desirable community according to $ci$. This may occur for instance, when a user is suspected of being a spy. By design, this conflict is resolved in G-SIR through *deny overrides*: if a user is classified as inhibitor, the access request is denied.

### 5.3 Geo-Social Obligations

Geo-social obligations establish that after activating a role, the requester needs to visit or cannot visit a particular place or interact with people within a predefined period of time.

**Definition 6.** *A* Geo-Social Obligation $b \in \mathbf{B}$ *is defined as* $\langle dir, D, \varphi \rangle$ *where*

- *dir is the directive that users subject to b need to fulfill.* $dir \in \{\langle +meet, \mathcal{S} \rangle, \langle +visit, SC \rangle, \langle -meet, \mathcal{S} \rangle, \langle -visit, SC \rangle\}$. *Here,* +meet, *means that the user should meet a targeted person or group as defined by social predicate $\mathcal{S}$, while* -meet *means that the user should not meet the person or population. Similarly,* +visit *means a user needs to visit spatial scope $SC$ and* -visit *that he cannot visit it.*
- *$D$ indicates the time duration when a user has to fulfill obligation b after b has been triggered and assigned to him.*
- $0 \leq \varphi \leq 1$ *is a value that represents how critical it is for the organization if a user violates the obligation. Here,* $\varphi = 1$ *means that it is very critical and* $\varphi = 0$ *means it is not critical at all.*

G-SIR instantiates each triggered obligation and monitors its state. Suppose user $u$ activates role $r$ and $b \in r.\mathcal{B}$. The framework creates a record that contains $u$, $b$, the time of activation $t$, and state of $b$, which can be *pending*, *fulfilled* or *violated*. The obligation should be fulfilled within the interval $[t,\ t + b.D]$. The obligation's state is *pending* when user $u$ has not fulfilled it and the deadline has not passed. The state changes to *fulfilled* if $u$ successfully fulfills $b$ and to *violated* if the user does not complete the required condition before $b.D$ elapses.

**Example 4.** *After activating a role, r, users may not enter the server room where the tenant's machines are stored and cannot meet people associated with community Y. Hence,* $r.\mathcal{B}$=$\{b_1, b_2\}$, *where* $b_1$=$\langle\langle -visit, \langle serverRoom, in \rangle\rangle, 1month, 0.7\rangle$ *and* $b_2$=$\langle\langle -meet, belongsToCommunity(u?, Y)\rangle, 1year, 0.5\rangle$.

### 5.4 Geo-Social Trace Constraints

Geo-social traces specify the locations and social interactions that are required before activating a role. When a user wants to activate a geo-social role, his traces are evaluated to see if they match the expected ones. If they do not match, the access request is denied.

**Definition 7.** *A* Geo-Social Trace Constraint $w \in \mathbf{W}$ *is a tuple* $w = \langle lst, D, \varphi \rangle$ *where*

- $lst = \langle\langle SC_1, \mathcal{S}_1\rangle_1, ... \langle SC_n, \mathcal{S}_n\rangle_n\rangle$ *is a list of places and/or people that the requester needs to visit and/or meet. $SC_i$ represents a spatial scope and $\mathcal{S}_i$ a social predicate that defines people that the requester needs to meet. When $SC_i$ or $\mathcal{S}_i$ is set to $\perp$, it indicates that the component needs no consideration.*
- *D is the duration that defines how long ago with respect to the current time in the recent past the trace should have been satisfied.*
- $0 \leq \varphi \leq 1$ *is the criticality associated with not completing the trace as expected.*

In the previous definition $w.D$ specifies that only recent traces are relevant. If a user is requesting access to a role that requires the fulfillment of $w.D$ at time $t$, the user should have completed the trace within $[t - w.D, t]$.

Recall that a single role may have one or more geo-social trace constraints; for a role $r$ the set of geo-social traces is denoted as $r.\mathcal{W}$. We use function *completeTraces(r, u)* to verify if $u$'s traces satisfy all the geo-social trace constraints $w \in r.\mathcal{W}$ associated with $r$.

**Example 5.** *Consider a medical doctor who is required to go to the Sanitizing Facility before entering into the Neo-natal Unit where new babies are born. This constraint can be expressed as* $w_1$=$(\langle\langle Sanitizing\ Facility, in\rangle, \perp\rangle, 15minutes, 0.8)$, *which is a trace constraint that requires the doctor to go to the Sanitizing Facility before being able to activate the role that allows him to enter into the Neo-natal Unit. If the user tries to gain access to a new-born unit without passing through the Sanitizing Facility, the impact of his actions may be severe because of the germs that he may be bringing to the newly born babies who are especially susceptible to infectious diseases. Hence, the criticality of the obligation is large,* $w_1.\varphi = 0.8$. *At the verification time, the system verifies that the requester completed the trace within the past 15 minutes. If he did not, completeTraces(r, u) returns false and the role cannot be activated.*

### 5.5 Well-Formed Policy

For G-SIR to work properly, it is necessary to ensure that the policy specification is consistent. Contracts are rules that forbid some interactions and movements; if they are violated access is denied. Hence, they should not conflict with any of the other constraints. Additionally, users should not be subjected to contradictory constraints. In Appendix A, we formally define the conditions to verify when a policy is *well-formed*. Whenever a new role is created or a user is assigned to a role, it is necessary to ensure that the resulting policy is well-formed according to those conditions.

### 5.6 Role Activation

With all the geo-social constraints specified, we now define how to make AC decisions in G-SIR.

**Definition 8.** *A role r with constraint vector* $CV_r = \langle SC, \mathcal{E}, \mathcal{I}, \mathcal{W}, \mathcal{GC}\rangle$ *is said to be* fulfilled *for user* $u_r$, $fulfilled(u_r, r)$, *iff the following conditions are satisfied:*

1) $r \in assigned(u_r)$
2) $validLocation(u_r, r)$
3) $completeTraces(r, u_r)$
4) $fulfillContracts(u_r)$
5) $inhibitors(u_r, r.\mathcal{I}) = \emptyset$
6) *If* $r.\mathcal{E} \neq \emptyset$, *then* $enablers(u_r, r) \neq \emptyset$

*Otherwise r is* not-fulfilled *for* $u_r$.

We now define how the system decides to grant or deny an access request $\mathbf{Q}_u$.

**Definition 9.** *An access request* $\mathbf{Q}_u = \langle u_r, P'\rangle$ *is granted under context* $\mathcal{X}_u$, if and only if *there exists a set of roles* $R' \subseteq R$ *such that all of the following conditions are fulfilled:*

1) $\bigcup_{r \in R'} P_{au}(r) \supseteq P'$ *(Roles in $R'$ provide the requested permissions),*
2) $\forall\ r \in R' :\ fulfilled(u_r, r)$ *(Definition 8)*
3) $RiskMan(\mathbf{Q}_u, \mathcal{X}_u) = true$

The last condition specifies that for $\mathbf{Q}_u$ to be granted, its associated risk should be acceptable according to function $RiskMan$. Next, we present how to compute $RiskMan$.

## 6 G-SIR Risk Management

In this section, we present how to compute $RiskMan$. Because *utility theory* has been recognized as a useful methodology to make decisions under uncertainty [21], we utilize it to formulate our decision-making process. A utility value represents the preferences of a decision maker. It is often useful to think of utility as a measure of *satisfaction*. A higher utility indicates a higher preference for an outcome and in combination, utility values reflect the preferred order of different outcomes. As it is customary, we define the utility value as a number between 0 to 100.

Consider an access request $\mathbf{Q}_u = \langle u_r, P' \rangle$ and let $R' \subseteq R$ be a set of roles that could satisfy the access request for $u_r$. We denote by $\mathtt{A}$ the uncertain event of $\mathbf{Q}_u$ being issued to compromise the system (an attack) and $\bar{\mathtt{A}}$ the complementary event ($\mathbf{Q}_u$ is a non-malicious request). We denote the probability of event $\mathtt{A}$ (attack) as $q$, hence the probability of event $\bar{\mathtt{A}}$ (no-attack) is $(1 - q)$. Similarly, let $\mathtt{G}$ represent allowing access and $\bar{\mathtt{G}}$ represents the decision to deny access.

The utility depends on the context of the user $\mathcal{X}_u$ and the permissions that $u_r$ would obtain through $R'$. There are four possible outcomes with corresponding utilities, of granting or denying $\mathbf{Q}_u$. These are: *i)* $\mathtt{U}^{R',\mathcal{X}_u}_{\bar{\mathtt{G}}/\mathtt{A}}$ is the utility of denying access to roles $R'$ under context $\mathcal{X}_u$ given that the access request is an attack, *ii)* $\mathtt{U}^{R',\mathcal{X}_u}_{\bar{\mathtt{G}}/\bar{\mathtt{A}}}$ is the utility of denying access to $R'$ under context $\mathcal{X}_u$ given that the access request is not an attack, *iii)* $\mathtt{U}^{R',\mathcal{X}_u}_{\mathtt{G}/\mathtt{A}}$ is the utility of granting access to $R'$ under context $\mathcal{X}_u$ given that the access request is an attack, and *iv)* $\mathtt{U}^{R',\mathcal{X}_u}_{\mathtt{G}/\bar{\mathtt{A}}}$ is the utility of granting the access when it is not an attack. Henceforth, we do not explicitly indicate the request being evaluated $\mathbf{Q}_u$, the set of roles $R'$ and current context $\mathcal{X}_u$ to simplify the notation. The expected utility (EU) of denying and granting access is computed as follows:

$$EU(\bar{\mathtt{G}}) = q * \mathtt{U}_{\bar{\mathtt{G}}/\mathtt{A}} + (1 - q) * \mathtt{U}_{\bar{\mathtt{G}}/\bar{\mathtt{A}}} \qquad (1)$$
$$EU(\mathtt{G}) = q * \mathtt{U}_{\mathtt{G}/\mathtt{A}} + (1 - q) * \mathtt{U}_{\mathtt{G}/\bar{\mathtt{A}}} \qquad (2)$$

An access request should be granted when $EU(\bar{\mathtt{G}}) \leq EU(\mathtt{G})$, otherwise it should be denied. Therefore, the threshold to decide when an access should be granted or denied can be computed by equalizing equations (1) and (2), where the only unknown value in the resulting equation is $q$. Solving for $q$, we find the threshold value for an access as per the following definition.

**Definition 10.** *Given the utility values* $\mathtt{U}_{\bar{\mathtt{G}}/\mathtt{A}}$, $\mathtt{U}_{\bar{\mathtt{G}}/\bar{\mathtt{A}}}$, $\mathtt{U}_{\mathtt{G}/\mathtt{A}}$ *and* $\mathtt{U}_{\mathtt{G}/\bar{\mathtt{A}}}$ *for context* $\mathcal{X}_u$*, request* $\mathbf{Q}_u$ *for which a set of roles* $R'$ *are enabled for* $u_r$*, the decision-making threshold is defined as follows:*

$$\tau(R', \mathcal{X}_u) = \frac{\mathtt{U}_{\mathtt{G}/\bar{\mathtt{A}}} - \mathtt{U}_{\bar{\mathtt{G}}/\bar{\mathtt{A}}}}{\mathtt{U}_{\bar{\mathtt{G}}/\mathtt{A}} + \mathtt{U}_{\mathtt{G}/\bar{\mathtt{A}}} - \mathtt{U}_{\mathtt{G}/\mathtt{A}} - \mathtt{U}_{\bar{\mathtt{G}}/\bar{\mathtt{A}}}}$$

*If* $\tau > 1$ *then* $\tau = 1$ *and if* $\tau < 0$ *then* $\tau = 0$.

The utility values depend on the context and request; hence, a different threshold is used for each context and request. The risk management procedure is defined as follows.

**Definition 11.** *Let* $R'$ *be a set of roles enabled for* $u_r$ *that satisfies request* $\mathbf{Q}_u$ *under context* $\mathcal{X}_u$*. The risk management decision-making process is as follows:*

$$RiskMan(R', \mathcal{X}_u) = \begin{cases} true & \textit{if } \tau(R', \mathcal{X}_u) > Pr[\mathtt{A} \mid \mathcal{X}_u, R'] \\ false & \textit{if } \tau(R', \mathcal{X}_u) \leq Pr[\mathtt{A} \mid \mathcal{X}_u, R'] \end{cases}$$

*where* $Pr[\mathtt{A} \mid \mathcal{X}_u, R']$ *is the probability of* $\mathbf{Q}_u$ *being an attack given* $\mathcal{X}_u$ *and* $R'$.

**Example 6.** *Consider a doctor trying to access a patient's record in two different contexts. Suppose that a set of roles* $R'$ *could satisfy the doctor's request,* $\mathbf{Q}_u$*. In context* $\mathcal{X}_{u1}$*, he is trying to access from an emergency room and in context* $\mathcal{X}_{u2}$ *he is requesting the same record from his home. The utility values for both contexts and the threshold values are presented in Table 2. Because the utility is a measure of satisfaction, the utility of granting access in an emergency room is larger than denying the access when the doctor is at home when there is no attack. This is true considering that granting access to a patient's data from the emergency room may save the patient's life. Similarly,* $\mathtt{U}^{R',\mathcal{X}_{u1}}_{\bar{\mathtt{G}}/\bar{\mathtt{A}}} < \mathtt{U}^{R',\mathcal{X}_{u2}}_{\bar{\mathtt{G}}/\bar{\mathtt{A}}}$*, because we would be less satisfied to have an access denied in the emergency room than in the other context. Given these utilities, the thresholds are computed according to Definition 10. Suppose* $Pr[\mathtt{A} \mid \mathcal{X}_{u1}, R']$=$Pr[\mathtt{A} \mid \mathcal{X}_{u2}, R']$=0.8*. In this case, we have* $\tau(R', \mathcal{X}_{u1})$=0.85 $>$ 0.8*, so the access is granted. This is equivalent to finding the expected utilities in equations (1) and (2), for which the analysis shows that* $EU(\mathtt{G}, \mathcal{X}_{u1})$=18 *and* $EU(\bar{\mathtt{G}}, \mathcal{X}_{u1})$=13*. Since the expected utility of granting is greater than the utility of denying the access, in this case, the best decision is to grant the access. In context* $\mathcal{X}_{u2}$*,* $\tau(R', \mathcal{X}_{u2})$=0.71 $<$ 0.8*, so the access is denied. Hence, when the access is from home,* $\mathcal{X}_{u2}$*, the system requires a larger assurance that the request is not an attack, whereas in a more critical type of access such as* $\mathcal{X}_{u1}$*, the system is more tolerable to the risk of attack because the associated utility values allow a riskier behavior.*

| | Utility for $\mathbf{Q}_u$, R′ | | | | |
|---|---|---|---|---|---|
| Context | $\mathtt{U}_{\mathtt{G}/\mathtt{A}}$ | $\mathtt{U}_{\mathtt{G}/\bar{\mathtt{A}}}$ | $\mathtt{U}_{\bar{\mathtt{G}}/\bar{\mathtt{A}}}$ | $\mathtt{U}_{\bar{\mathtt{G}}/\mathtt{A}}$ | $\tau$ |
| $\mathcal{X}_{u1}$ (emergency room) | 0 | 90 | 5 | 15 | 0.85 |
| $\mathcal{X}_{u2}$ (remote access) | 0 | 70 | 10 | 25 | 0.71 |

TABLE 2
Example utility values for two different contexts.

**Obtaining Utility Values:** Utility values are subjective in nature and, therefore, each organization should elicit them. An in-depth review of the widely studied *utility elicitation process* can be found in [21]. Utility values should satisfy the following relations to be correct. First, $\mathtt{U}_{\mathtt{G}/\mathtt{A}} < \mathtt{U}_{\mathtt{G}/\bar{\mathtt{A}}}$, as an organization would be clearly more satisfied if an access request is granted and it turns out to be a legitimate access request, than if granting access results in an attack. Similarly, $\mathtt{U}_{\bar{\mathtt{G}}/\mathtt{A}} < \mathtt{U}_{\bar{\mathtt{G}}/\bar{\mathtt{A}}}$, because an organization is more satisfied if an access request issued to attack the organization is denied than if a non-malicious access is denied.

## 7 Enforcement Algorithm

To enforce the G-SIR policy, we propose Algorithm 1. The inputs to the algorithm are the requester $u_r$, a set of requested permissions $P'$, the location of the requester $\mathcal{L}_u$ and his context $\mathcal{X}_u$. The algorithm looks for a set of roles $R'$ to satisfy the access request. If at the end of the execution $R'$ is empty, the access is denied. Otherwise, it is granted. Next, we describe the working of the algorithm.

Candidate role selection: First, the algorithm verifies if the requester $u_r$ is violating any contract in line 2, and if he is, the access is denied. Next, in line 4 the set of candidate roles $R_{avail}$ is found using function *getCandidateRoles* (presented in line 11). In line 12, the function verifies if all permissions in $P'$ can be obtained through the roles assigned to $u_r$. If not, the request cannot be granted because there are no roles assigned to $u_r$ that provide $P'$. In which case,

**Algorithm 1** Geo-Social Decision Making Process

**Input:** $u_r$:= requesting user, $P'$:= Permissions requested, $\mathcal{L}_u$:= location of $u_r$, $\mathcal{X}_u$:= context of $u_r$.
**Output:** $R'$:= set of roles that fulfill Definition 9. If $R' \neq \emptyset$, the access is denied. Otherwise, roles $R'$ can be activated to grant the access request.

```
 1: findGeoSocialRoleActivationSet($u_r, P', \mathcal{L}_u, \mathcal{X}_u$)
 2: if ¬fulfillContracts($u_r$) then
 3:    return $\emptyset$ {Request denied}
 4: $R_{avail} \leftarrow$ getCandidateRoles($u_r, P', \mathcal{L}_u$) {Candidate roles}
 5: if $R_{avail} = \emptyset$ then
 6:    return $\emptyset$ {Request denied}
 7: $R_{sel} \leftarrow \emptyset$ {Selected roles so far}
 8: $P_{rem} \leftarrow P'$ {Set of permissions that haven't been found}
 9: $R' \leftarrow selectRolesMinimumRisk(P_{rem}, R_{avail}, R_{sel}, u_r, \mathcal{X}_u)$
10: ————————————————————————————————
11: getCandidateRoles($u_r, P', \mathcal{L}_u$)
12: if $(P' \setminus P_{au}(assigned(u_r))) \neq \emptyset$ then
13:    return $\emptyset$ {Authorized roles cannot provides $P'$}
14: $R_{avail}, R_i \leftarrow \emptyset$
15: for all $r \in assigned(u_r)$ do
16:    if $(P_{au}(r) \cap P' \neq \emptyset)$ then
17:       $R_i \leftarrow R_i \cup \{r\}$
18: for all $r \in R_i$ do
19:    if validLocation($u_r, r$) ∧ completeTraces($r, u_r$)
       ∧ (inhibitors($u_r, r$) = $\emptyset$) then
20:       if enoughNonColludingEnablers($r, u_r$) then
21:          $R_{avail} \leftarrow R_{avail} \cup \{r\}$
22: if $(P' \setminus P_{au}(R_{avail})) \neq \emptyset$ then
23:    return $\emptyset$ {Roles in $R_{avail}$ cannot provides $P'$}
24: return $R_{avail}$
25: ————————————————————————————————
26: enoughNonColludingEnablers($r, u_r$)
27: for all $ce \in r.\mathcal{E}$ do
28:    $U_{avail} \leftarrow \emptyset$
29:    $U_v \leftarrow$ vicinity($ce.SC$) $\setminus u_r$
30:    if $|U_v| < ce.k$ then
31:       return false
32:    for all $u_v \in U_v$ do
33:       if fulfillContracts($u_v$)
          ∧ fulfillSocialPredicate($u_r, u_v, ce.\mathcal{S}$) then
34:          $U_{avail} \leftarrow U_{avail} \cup \{u_v\}$
35:    if $ce.k \leq |U_{avail}|$ then
36:       $found \leftarrow false$
37:       while $U_a \subseteq combinations(U_{avail}, ce.k) \wedge \neg found$ do
38:          if PrCollusion($U_a \cup \{u_r\}$) $< ce.\tau$ then
39:             $found \leftarrow true$
40:       if $\neg found$ then
41:          return false {Couldn't find enablers for $ce$}
42:    else
43:       return false {Not enough users in $U_{avail}$}
44: return true {All enabling constraints are satisfied.}
```

an empty set of available roles is returned and the access is denied. Otherwise, the function continues its execution initializing variables $R_{avail}$ and $R_i$. $R_{avail}$ is a set used to store roles assigned to $u_r$ that have all its constraint vectors fulfilled according to Definition 8. $R_i$ is a set variable used to store roles that provide one or more permissions in $P'$. Both $R_{avail}$ and $R_i$ are initially empty. In line 15, all roles assigned to $u_r$ are evaluated and only those that provide requested permissions are added to $R_i$. Then, in line 18 all roles in $R_i$ are verified to see if their constraint vectors are satisfied. This verification consists of evaluating the following conditions (line 19): that $u_r$'s current location allows the activation of $r$, that $u_r$ has completed the traces required for the activation of $r$ and that there are no users in the vicinity who conflict with $r$'s inhibiting constraint. If these conditions are satisfied, the function proceeds to evaluate if the enabling constraints associated with $r$ are also fulfilled. For this purpose, in line 20 a function that verifies $r$'s enabling constraints is invoked (we discuss this function later). If $r$'s constraint vector is satisfied, $r$ is added to $R_{avail}$ in line 21. Hence, $R_{avail}$ only contains roles with fulfilled constraint vector. Because $R_{avail}$ may be a subset of $R_i$, one last verification is performed. In line 22, roles in $R_{avail}$ are verified to see if they can provide all the permissions in $P'$. If they cannot, the function returns an empty set and the access is denied. Otherwise, $R_{avail}$ is returned in line 24.

Finding non-colluding enablers: To find the set of non-colluding enablers function *enoughNonCollidingEnablers* is invoked in line 20. This function is presented in line 26. Variable $U_{avail}$ is initially empty and is used to store users who are potential enablers. For each enabling constraint $ce$ associated with role $r$ (line 27), the set of users in the vicinity are retrieved and stored in $U_v$ (line 29). Users in $U_v$ are examined to determine if they are violating their contracts or do not fulfill the required social predicate (line 33). Only users who are not violating their contracts and fulfill $ce$'s social predicate are added to $U_{avail}$. After that, if $U_{avail}$ does not have the required $ce.k$ the function returns false to show that there are no valid enablers for $r$ (line 43). Otherwise, groups of size $k$ are evaluated in line 37. If none of the groups evaluated are collusion free, the function returns *false* to show that there are no valid enablers for $ce$. If a group $U_a$ is found to be non-colluding with the required probability, $ce$ is satisfied and variable *found* is set to true to show that there is no need to continue examining other groups. It is necessary to ensure that all enabling constraints in $r.\mathcal{E}$ are satisfied; hence, the function continues evaluating all constraints (for loop line 27). If after all constraints $ce \in r.\mathcal{E}$ have been evaluated and a set of collusion free enablers has been found for each $ce$, the function returns *true* in line 44.

Selection of roles to activate with minimum risk: After $R_{avail}$ is found, it is guaranteed to uniquely have roles assigned to $u_r$ for which corresponding constraint vectors are fulfilled. If $R_{avail}$ is empty, there are no roles and the access is denied (line 6). Otherwise, the algorithm proceeds to find the roles to be activated. There may be multiple subsets of roles in $R_{avail}$ that could satisfy the request. To select the set of roles to be activated, we use the risk exposure function in Definition 11 and, in line 9, invoke our previously proposed algorithm presented in [22], which selects the set of roles that minimizes the risk exposure. After function *selectRolesMinimumRisk* is invoked, it returns the set of roles with minimum risk exposure that can be activated by $u_r$ to satisfy the request. If the function returns an empty set, no role can be activated to satisfy the access request and the access is denied. Otherwise, the access is granted by activating $R'$.

## 8 Experimental Evaluation

We evaluate the proposed system using a discrete indoor simulator implemented in Java. We describe the experimental setup and then the experimental results. Our dataset can be download from https://github.com/NathalieB1/G-SIR.git

### 8.1 Experiment Setup

**Generation of social graph and user mobility:** For simulating user mobility, we randomly generated a map where the assumed organization is located, as follows. First, we specified a size of a Cartesian rectangle. Then, we randomly selected the points where places are located on the map. These places were also randomly connected according to the parameters specified in Table 3. In our implementation, we used a graph abstraction where vertices represent the places on the map and edges represent connections between places (e.g., corridors). At the beginning of the simulation, all users were randomly placed on the map. Each policy was evaluated at multiple time instants, and at every time instance users could move around the map to adjacent places or stay in their current positions.

Social graphs were generated using the Jung API provided in [23]. We evaluated the effect of representative types of network topologies on the system. We evaluated topologies commonly observed in social networks according to [24]: *preferential attachment*, *small world*, *power law* and a fully connected network. All graphs evaluated were undirected.

**Generation of policy and access requests:** Policies were randomly generated using the parameters presented in Table 3. We ensured that all policies used in the experiments were well-formed according to the Appendix. We selected the values of the parameters inspired by previous works such as [10], [25], which evaluate RBAC policy enforcement. To the best of our knowledge, no earlier work includes the evaluation of geo-social policies. We adjusted some parameters and included new ones to incorporate unique geo-social features. In this section, we test different values for those parameters to show their effect. Roles' activation thresholds (which represent the maximum tolerable probability of attack, Definition 10) were randomly assigned between 0 and 0.5 to represent that the information accessed through those roles is valuable. The probabilities of attack used for the risk management procedure were randomly generated for each user. We evaluated the effect of the estimation error in one experiment. Throughout the simulation, the probabilities of attack for each user were randomly updated. Initially, the probabilities of attack were set to 0.01.

To generate inhibiting constraints, we created three classes of confidential data and assigned to each class a color that represents the type of individuals who should not be allowed to access it. When a role was generated with an inhibiting constraint, one color was randomly selected. At the beginning of the simulation, we randomly selected inhibiting users and tainted them with a random color.

Enabling constraints were randomly generated to require $k$ users related by friendship to the requester in the spatial scope of the role to be activated. Colluding communities were randomly generated. We considered two parameters, the number of colluding communities and the number of members per each colluding community. For each community, we randomly selected a user and marked him as colluding and then, continued choosing some of his friends as colluding until the number of colluding users per community was reached.

Trace constraints were generated randomly verifying that the path required to arrive at the place of access did indeed exist. The number of previous places users needed to visit was set as 2. The time required to fulfill the constraint (Definition 7) was generated considering the distance between places and the speed of users to allow enough time.

**Request generation, events counted as threats and improvement measure:** Every time a user stepped into a place where there was a role with spatial scope, an access request was issued on his behalf. We consider the following as potential insider threats. *i) Unauthorized for role:* A user tries to assume a role he is not authorized for. *ii) Inhibiting users:* A user issues an access request, but there are inhibiting users in the vicinity that may launch a proximity attack, e.g., if an access request is evaluated for a role with color *red* and a user tainted *red* is in the vicinity, he is classified as inhibitor. *iii) Lack of enablers:* A request is issued, but there are not enough enablers at the required place to authorize the request. *iv) Colluding users:* A user issues an access request that requires enablers and the only people who could serve as enablers are colluding according to the collusion threshold. *v) Enablers violating contracts:* A user issues a request for which all potential enablers are violating their contracts (they are in places where they should not be). *vi) Suspicious requester:* A user issues a request for which the probability of attack is too high compared to the role's activation threshold. *vii) Incomplete traces:* A user issues a request without completing the required traces. Each access request was only classified in a single category according to the order of constraint evaluation in Algorithm 1.

| Parameter | Value |
|---|---|
| Ratio of number of places to number of users | 1:3 |
| User speed | 5 ft/second |
| Map coordinates (size of Cartesian map) | (300 ft x 300 ft) |
| Ratio of number of users to roles | 4:1 |
| Ratio of roles assigned per user to num. of roles | 1:2 |
| Inhibiting constraints per role | 1 |
| Percentage of roles with inhibiting constraints | 50% |
| Types of inhibiting users (colors) | 3 |
| Percentage of inhibiting users | 40% |
| Enabling constraints per role | 1 |
| Range of $k$ | [1, 3] |
| Required social relation | Friendship |
| Collusion threshold | 0.9 |
| Number of colluding communities | 5% of users |
| Number of colluding users per community | 5 |
| Roles with trace-based constraints | 5% |
| Roles with geo-social contracts | 40% |
| Simulation time | 8 hours |

TABLE 3
Default experimental parameters. The number of users is used as *policy size*.

To evaluate our proposed approach, we use as a baseline the most similar approach: the Geo-Social RBAC model [9]. We measure the percentage of threats detected by G-SIR in comparison to the baseline; refer to this number as *improvement* and it is computed as follows: $improvement = \left[\frac{n_{proposed}}{n_{baseline}}\right] - 1$, where $n_{proposed}$ is the number of potentially malicious access requests (threats) detected by G-SIR which is equal to the sum of all previously described threats and $n_{baseline}$ is the number of threats detected by the baseline, which is the addition of threats of type *(i)*, *(iii)* and *(vii)*. In the following, we show the improvement as a percentage.

## 8.2 Analysis of Results

In this subsection, we discuss and analyze the results of our various experiments. Each reported experimental measurement is the average of running the simulation 30 times (each time a different randomly generated policy was used). The results presented were found using 30 randomly generated OSNs, 10 of each topology. The number users indicates the *policy size*. We vary the policy size in the experiments to test G-SIR' scalability. The default number of users in the simulation were fixed at 250.

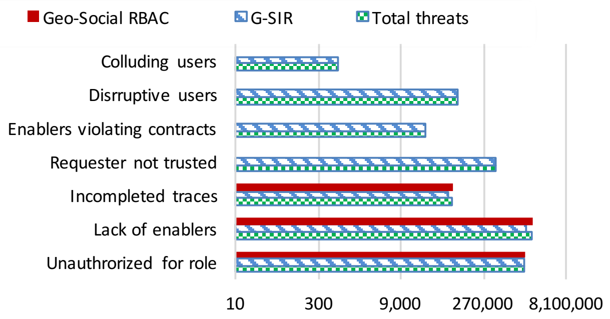


Fig. 2. Comparison between the number of threats prevented by G-SIR and that by the baseline (Geo-Social RBAC). Plot in logarithmic scale

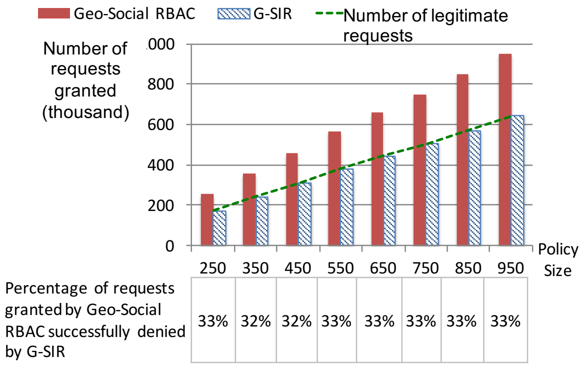


Fig. 3. Comparison between the number of requests granted by G-SIR and that by the baseline (Geo-Social RBAC).

**Baseline Comparisons:** First, we present an overview of the types of attacks prevented by the proposed G-SIR with respect to the baseline, Geo-Social RBAC. Figure 2 shows the number of threats detected by G-SIR that could not be captured by the baseline. The number of access requests denied because the requester was unauthorized for a role; hence, this number is the same for both approaches. The number of requests denied due to *lack of enablers* and *incomplete traces*, is slightly larger for the baseline, because G-SIR finds other policy violations first, according to the order presented in Algorithm 1. G-SIR captures these four additional types of violations that are not considered by the baseline: *suspicious requester*, *enablers violating contracts*, *inhibiting users* and *colluding users*. There may be some misclassifications in the counts of *suspicious requester* and *colluding users*. This is caused by the uncertainty in the estimation of the probability of attack and the probability of collusion, respectively. In a later experiment, we present the effect of the number of false positives and false negatives.

Figure 3 presents the number of requests granted by the baseline and our proposed approach. In the x-axis, we show the results for varying policy sizes. The dotted line represents the number of requests granted that are legitimate. All requests that are above that line are malicious ones and should not have been granted. The table below the figure contains the exact percentage of malicious requests granted by the baseline that G-SIR was successfully able to deny. Overall, the results show that the baseline granted around 33% of malicious requests irrespective of the policy size. The policy size uniquely influenced the total number of requests granted. Overall, G-SIR was able to identify 33% more policy violations than the baseline.

The difference between the percentage of threats captured in Figure 3 and the malicious requests granted by Geo-Social RBAC in 3 is due to the difference between the number of requests generated by the simulator that were granted and the number of requests that were denied. The number of requests denied due to the lack of assigned role, and lack of enablers is substantially larger than all other types of requests (Figure 2) including the number of granted requests. Hence, the improvement reported is greater than when only the number of granted requests is considered.

The percentage of additional threats captured by G-SIR depends on the type of policy enforced. In the following, we present the effect of increasing both the percentage of attacks that are addressed by G-SIR and the different types of constraints in the system. Unless explicitly mentioned, parameters are maintained to their default values (Table 3).

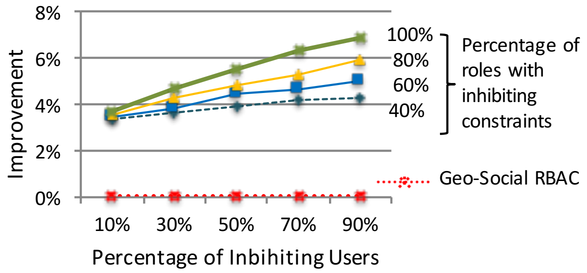


Fig. 4. Effect of inhibitng users and different ratios of roles with inhibiting constraints.

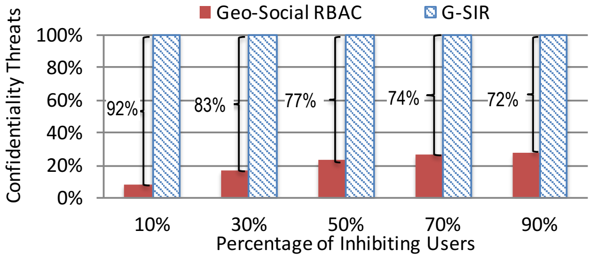
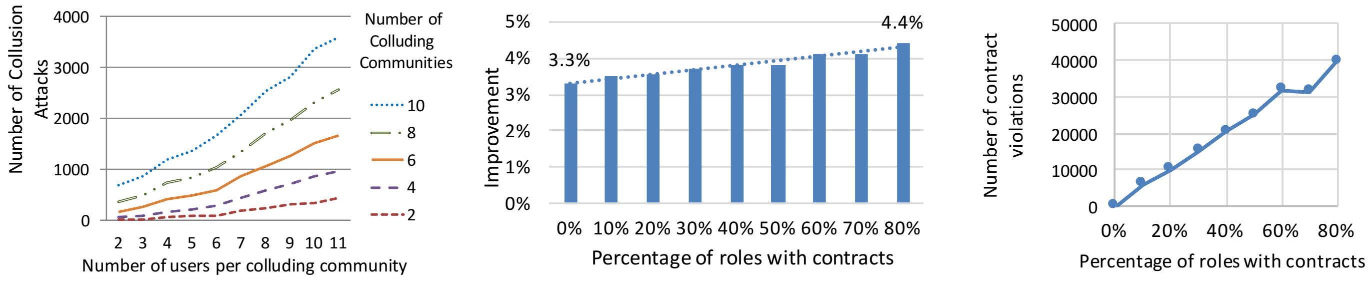


Fig. 5. Percentage of confidentiality threats prevented.

**Proximity Attacks:** Figure 4 presents the effect of environments with varying number of inhibiting users under policies with various number of roles with inhibiting constraints. In this figure, the baseline is represented by the line in 0%. As the number of roles with inhibiting constraints increases, there are more confidentiality leaks prevented. Similarly, as the number of inhibitors in the vicinity increases, the number of leaks of confidential information due to proximity attacks is also higher. Since the baseline does not prevent this type of attacks, the overall number of threats prevented by G-SIR increases. For policies where 80% of roles had inhibiting constraints and when only 10% of the users were inhibitors, G-SIR was able to capture 3.5% more threats than the baseline. For systems where 90% of the users cannot learn some information (recall that there are three colors), the improvement was 5.9%. When 100% of users are assigned an inhibiting color and every role has an inhibiting constraint associated with it, the number of threats mitigated goes up to 6.8%.

In our next experiment, we counted as a confidentiality threat an attempt to do any of the following: *i)* when a user



(a) Collusion threats captured by G-SIR.

(b) Effect of contract enforcement in the overall threat detection.

(c) Contract violations detected by G-SIR.

Fig. 6. Effect of collusion attacks and geo-social contracts on the number of threats captured.

tries to assume a role that he is not assigned to and the role has an inhibiting constraint that conflicts with the color of the user, *ii)* when there are inhibitors in the vicinity, *iii)* when there is a collusion to access confidential information and *iv)* when there is a contract violation and the violating user is trying to serve as an enabler. Using this classification, in Figure 5, we present a comparison between the percentage of confidentiality threats detected by G-SIR in contrast to Geo-Social RBAC. In this experiment, the proposed G-SIR captures more confidentiality threats than those captured by the baseline (Geo-Social RBAC). Geo-Social RBAC only captures confidentiality threats of type *i)*. As the percentage of inhibiting users increases, the number of all types of confidentiality violation attempts enumerated before also increases; including those of type *i)*. That is why we see that the percentage of confidentiality threats captured by Geo-Social RBAC does increase with the number of inhibiting users. However, there is always a large percentage of threats that are not detected by Geo-Social RBAC. Figure 5 was generated for policies where 60% of roles have inhibiting constraints. For these policies, the percentage of threats not captured by Geo-Social RBAC vary between 92% and 72%. For policies with a higher number of roles with inhibiting constraints, the number of threats not captured by Geo-Social RBAC that are captured by our G-SIR is larger. For instance, when all roles have inhibiting constraints and there are 90% of inhibiting users, the percentage of threats not captured by the baseline captured by G-SIR increases to 76%. This corresponds to 4% more than the same data point in Figure 5. These experiments show that G-SIR is effective in capturing proximity and confidentiality threats.

**Collusion Attacks:** Figure 6a presents the effect of increasing the number of colluding users per community (x-axis) and the number of colluding communities. Colluding attacks are not prevented by the baseline, hence all the lines in the figure represent attacks thwarted by G-SIR. The results reported were generated for policies with enabling constraints that required one enabler ($k$=1). In this experiment, we assumed that the colluding communities and users were known (in another experiment we show the effect of the accuracy of community detection algorithms). Hence, all attacks presented in Figure 6a can be prevented by G-SIR. Figure 6a shows that the number of colluding attacks prevented by G-SIR increases with the number of communities. This follows because the probability of detecting an attack when more communities exist is larger. Similarly, as the number of colluding members per community increases, the probability of a collusion attack increases and the number of collusion threats increases.

**Geo-Social Contract Violations:** Figure 6 presents the number of contract violations stopped by G-SIR as the percentage of roles with contract constraints increases. The baseline does not prevent any of these attacks. In Figure 6b, we present the overall increase on the number of threats uniquely prevented by G-SIR and in Figure 6c the number of contract violations. In both figures, as the number of geo-social contracts in the policy increases, the threats captured by the G-SIR also increases. This implies that for organizations that require more protection against users wandering through unauthorized places, G-SIR performs better. Recall that policies randomly generated by our simulator are well-formed and every role has a spatial scope assigned to it. Hence, the number of contract violations uniquely captures threats where a potential enabler is violating a contract. Had we used policies that contain conflicts, the number of violations reported would be larger, as the verification in line 2 of Algorithm 1 never evaluated to true during our simulations. Therefore, these figures uniquely show attacks that aim at using enablers that are not qualified to be in the required spatial scope. These figures show a clear trend where the number of attacks stopped increases as the roles with contracts is incremented. The fluctuations shown reflect users' random movements.

**Sensitivity and specificity analysis:** G-SIR takes as input the estimated probability of attack. In this experiment, we measure the effect of using estimation methodologies with different values of average error, $\varepsilon$, on the number of threats detected by G-SIR. We generated synthetic data as follows. We randomly selected a probability of attack, $q$, for each user; this value was considered as the ground truth. Then, the *estimated probability*, $\hat{q}$, was randomly selected in the interval $[q-\varepsilon/2, q+\varepsilon/2]$. We changed the value of $\varepsilon$ between 0.1 and 0.8. The observations generated by the simulation runs were classified as true positives (TP), false negatives (FN), false positives (FP) and true negatives (TN).

Figure 7 presents the results of this experiment, which include the average number of TP, FN, FP and TN as well as the average sensitivity and specificity. Sensitivity and specificity are measures that provide an overview of the relation between TP, FN, FP and TN. Sensitivity represents the percentage of attackers who are correctly identified as attackers while specificity shows the percentage of legiti-

| Estimation Error | Average Number TP | FN | FP | TN |
|---|---|---|---|---|
| 0.1 | 123726.1 | 0.7 | 252.7 | 5546.2 |
| 0.2 | 123725.6 | 1.3 | 444.5 | 5354.4 |
| 0.3 | 123724.8 | 2.0 | 656.0 | 5142.9 |
| 0.4 | 123724.7 | 2.1 | 887.3 | 4911.6 |
| 0.5 | 123723.4 | 3.4 | 1098.6 | 4700.3 |
| 0.6 | 123723.0 | 3.9 | 1265.8 | 4533.1 |
| 0.7 | 123721.3 | 5.5 | 1407.5 | 4391.4 |
| 0.8 | 123721.4 | 5.5 | 1537.7 | 4261.2 |

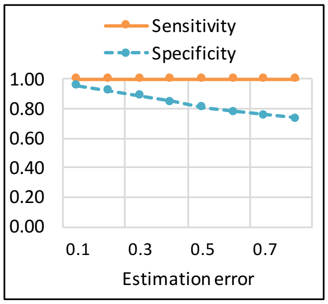


Fig. 7. Effect of the estimation error, $\varepsilon$, of the inference technique used on the number of threats captured by G-SIR.

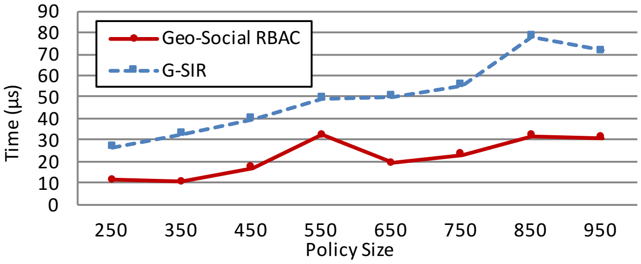


Fig. 8. Average time as the policy size increases in microseconds.

mate insiders who are correctly identified as not being a threat. The sensitivity measure shows that G-SIR is good in capturing attacks even when the estimation error increases. This is a consequence of the following. First, some of the policy constraints that are part of G-SIR do not depend on the inferred input data, so they can be enforced correctly without any influence of the estimation error. Secondly, the thresholds used in the simulation were selected to ensure that, as in real policies, only relevant information would be protected by G-SIR policies. Thus, when the inferred probability of attack is too high, even under certain error, the enforcement mechanism will deny access to the most important information. This can be seen in the average number of TP, which decreases very little as the estimation error increases. In the worst case, when $\varepsilon$=0.8, the number of TP is reduced on average by four observations which is relatively small compared to the total number of observations. The specificity shows that as the estimation error increases, the number of honest insiders who are denied access to very critical resources increases as well. That is, the effect of the estimation error can be seen on the average FP. These results indicate that G-SIR is capable of stopping most threats generated by the simulator even when the performance of the information module is not good.

**Runtime overhead:** In Figure 8, the difference between the time required by Geo-Social RBAC and G-SIR is shown for policies of multiple sizes. Our proposed G-SIR introduces some additional runtime overhead due to the extra verifications performed. However, the overhead is acceptable in comparison to Geo-Social RBAC.

## 9 Related Work

Several approaches have extended RBAC to include the context of users such as the location and temporal constraints as part of the AC decision [26], [27], [28], [29], [30]. However, they are not designed to incorporate the social dimensions of users and are not capable of adapting to negative changes in users' behavior. The explicit use of geo-social context as part of the AC policy is very recent [7], [8], [9]. However, these approaches were not designed to prevent insider threats and as a consequence, the risk exposure of geo-social threats is not mitigated. Prox-RBAC model [7] extends the Geo-RBAC model [26] by including proximity between individuals as part of the policy in indoor environments. Gupta *et. al* [8] extended Prox-RBAC by providing formal definitions for determining the proximity between locations, users, attributes and time. This type of predicates can be used by our model to define spatial scopes. Baracaldo *et. al* proposed Geo-Social RBAC in [9] which is the closest related work. Geo-Social RBAC includes geo-social trace based and cardinality constraints. We extended these two constraints by including a criticality value that serves to detect suspicious geo-social behavior. In Section 8, we compared our proposed approach with [9] and showed that G-SIR is capable of deterring substantially more insider threats than Geo-Social RBAC. Previous work was not designed to consider the risk of geo-social interactions to mitigate insider threats and do not include the use of geo-social obligations, contracts or vicinity constraints.

Other related work include adaptive AC systems such as [10], [11], [12], [13], [14], [15]. These approaches aim to revoke accesses when users are not behaving properly. In this type of AC systems, the behavior of the user is incorporated into the AC decision process by considering a trust level of users that is computed based on users' behavior. Nonetheless, these approaches were not designed to include geo-social information and are not capable of mitigating some of the threats presented in Section 2. We leveraged our previous approach presented in [10], where an optimization problem and an algorithm to select a set of roles that minimize the risk exposure of granting a request were defined. We incorporated that algorithm as part of G-SIR. There are several differences between the proposed solution and those in [10]. For most, the model presented in [10] does not consider the geo-social aspects of the requester. Additionally, in this paper, we presented a utility-based methodology to compute the threshold to allow an access while the methodology presented in [10] only considers the damage of granting an access, without including the expected gain of granting an access.

## 10 Conclusion

Little work exists in geo-social AC area and existing ones do not consider the intricacies of incorporating geo-social information as part of the AC system for insider threat mitigation. We performed an analysis of insider threats that arise when geo-social information is used to perform AC decisions. To capture these new threats, in this paper, we presented Geo-Social Insider Threat Resilient Access Control Framework (G-SIR). To the best of our knowledge, this is the first effort to use geo-social information to deter insider threats by incorporating it into the AC mechanism. We proposed an AC methodology that includes geo-social constraints and presented several geo-social constraints that include geo-social contracts, geo-social risk aware trace constraints, collusion free enabler constraints, inhibiting constraints and geo-social obligations. Enforcing these constraints help reduce the risk of proximity, social engineering and probing threats. Additionally, monitoring the fulfillment of these constraints helps identify suspicious users who are more prone to violate policy by visiting more than usual places where they should not be at. We provided an enforcement algorithm and presented simulation results to evaluate the proposed framework. Our experimental results show that the proposed approach is effective, scalable and deter insider threats.

G-SIR assumes it is possible to reliably monitor users' geo-social context. New methodologies are needed to systematically deploy reliable and tamper-proof indoor location systems, where users trying to manipulate their reported locations can be identified and flagged as suspicious. This requires orchestrating multiple location technologies, such as the ones presented in [31], to arrive at a cost-effective deployment solution.

## Acknowledgment

Joshi has been supported by NSA Cybersecurity grant–BAA-003-15.

## References

[1] A. Moore, D. Cappelli, and T. R., "The "big picture" of insider it sabotage across u.s. critical infrastructures," http://www.cert.org/insider_threat, 2008.

[2] CERT, "2014 us state of cybercrime survey," 2014.

[3] Y.-S. Cho, A. Galstyan, P. J. Brantingham, and G. Tita, "Latent self-exciting point process model for spatial-temporal networks," 2013.

[4] J. Kubica, A. Moore, J. Schneider, and Y. Yang, "Stochastic link and group detection," in *Proc. of the National Conf. on AI*, 2002.

[5] J. Adibi, H. Chalupsky, E. Melz, A. Valente *et al.*, "The kojak group finder: Connecting the dots via integrated knowledge-based and statistical reasoning," in *Proc. of the National Conf. on AI*, 2004.

[6] C. D. Corley, D. J. Cook, A. R. Mikler, and K. P. Singh, "Text and structural data mining of influenza mentions in web and social media," *Int. journal of environmental research and public health*, 2010.

[7] 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 Inf. Systems*, 2011.

[8] A. Gupta, M. S. Kirkpatrick, and E. Bertino, "A formal proximity model for rbac systems," *Computers & Security*, 2013.

[9] N. Baracaldo, B. Palanisamy, and J. Joshi, "Geo-social-rbac: A location-based socially aware access control framework," in *In Proc. of the 8th Int. Conf. on Network and System Security*, 2014.

[10] N. Baracaldo and J. Joshi, "A trust-and-risk aware rbac framework: tackling insider threat," in *Proc. of the 17th ACM SACMAT*, 2012.

[11] S. Chakraborty and I. Ray, "Trustbac: integrating trust relationships into the rbac model for access control in open systems," in *Proc. of the 11th ACM SACMAT*, 2006.

[12] F. Feng, C. Lin, D. Peng, and J. Li, "A trust and context based access control model for distributed systems," in *Proc. of the 10th IEEE Int. Conf. on High Perform. Comp. and Communications*, 2008.

[13] N. Dimmock, A. Belokosztolszki, D. Eyers, J. Bacon, and K. Moody, "Using trust and risk in role-based access control policies," in *In Proc. of the 9th ACM SACMAT*, 2004.

[14] B. Aziz, S. N. Foley, J. Herbert, and G. Swart, "Reconfiguring role based access control policies using risk semantics," in *Journal of High Speed Networks: Special Issue on Managing Security Policies, Modelling Verification and Configuration*, 2006.

[15] L. Chen and J. Crampton, "Risk-aware role-based access control," in *Proc. of the 7th Int. Workshop on Sec. and Trust Management.*, 2011.

[16] D. Boucher and P. Kelly, *The social contract from Hobbes to Rawls*, 2003.

[17] "Opengis simple features specification for sql, technical report ogc 99-049," OpenGIS Consortium, Tech. Rep., 1999.

[18] Q. M. S. Osborn, R. Sandhu, "Configuring role-based access control to enforce mandatory and discretionary access control policies," in *ACM Trans. on Inf. and System Security*, 2000.

[19] G. K. Palshikar and M. M. Apte, "Collusion set detection using graph clustering," *Data Mining and Knowledge Discovery*, 2008.

[20] A. S. Sabau *et al.*, "Survey of clustering based financial fraud detection research," *Informatica Economica*, 2012.

[21] R. Clement, "Chapter 13: Risk attitudes, utility function assessment," in *Making Hard Decisions, An Introduction to Decision Analysis*, 2nd ed., 1995.

[22] N. Baracaldo and J. Joshi, "An adaptive risk management and access control framework to mitigate insider threats," *Computers & Security*, 2013.

[23] "Jung: Java universal network/graph framework," http://jung.sourceforge.net, 2015.

[24] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proc. of the 7th ACM SIGCOMM*, 2007.

[25] G. T. Wickramaarachchi, W. H. Qardaji, and N. Li, "An efficient framework for user authorization queries in rbac systems," in *Proc. of the 14th ACM SACMAT*, 2009.

[26] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca, "Geo-rbac: a spatially aware rbac," in *Proc. of the tenth ACM SACMAT*, 2005.

[27] S. M. Chandran and J. B. Joshi, "Lot-rbac: A location and time-based rbac model," in *Web Information Systems Engineering*, 2005.

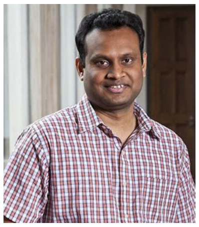
[28] M. Toahchoodee, I. Ray, and R. M. McConnell, "Using graph theory to represent a spatio-temporal role-based access control model." *Int. Journal of Next-Generation Computing*, 2010.

[29] 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 ACM SACMAT*, 2001.

[30] I. Ray, M. Kumar, and L. Yu, "Lrbac: a location-aware role-based access control model," in *Information Systems Security*, 2006.

[31] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," in *Proc. of the 14th int. conf. on inf. processing in sensor networks*, 2015.

**Nathalie Baracaldo** is a Research Staff Member at the IBM's Almaden Research Center in San Jose, CA. She received her Ph.D. from the School of Information Sciences at the University of Pittsburgh in 2016. She earned her M.S. Degree in Computer Sciences *(Cum Laude)* from Universidad de los Andes in Bogota, Colombia. Prior to that, she earned two undergraduate degrees in Computer Science and Industrial Engineering at the same university. Her research interests lie at the intersection of information security, privacy and trust. Some of the topics that she has explored are secure cloud storage systems, secure interoperability in distributed systems, risk management and mitigation of insider threats.



**Balaji Palanisamy** is an Assistant Professor in the School of Information Science in University of Pittsburgh. He received his M.S and Ph.D. degrees in Computer Science from the college of Computing at Georgia Tech in 2009 and 2013 respectively. His primary research interests lie in scalable and privacy-conscious resource management for large-scale Distributed and Mobile Systems. At University of Pittsburgh, he codirects research in the Laboratory of Research and Education on Security Assured Information Systems (LERSAIS), which is one of the first group of NSA/DHS designated Centers of Academic Excellence in Information Assurance Education and Research (CAE &CAE-R). He is a member of the IEEE.

**James Joshi** is a Professor of School of Information Sciences (SIS) at the University of Pittsburgh. He received his MS in Computer Science and PhD in Computer Engineering degrees from Purdue University in 1998 and 2003, respectively. He is an elected Fellow of the Society of Information Reuse and Integration (SIRI) and is a Senior member of the IEEE and the ACM. His research interests include Access Control Models, Security and Privacy of Distributed Systems, Trust Management and Information Survivability. He is the director of LERSAIS at the University of Pittsburgh.