Relationship-Based Access Control: Protection Model and Policy Language

UC CPSC Techical Report 2010-974-23

Philip W. L. Fong
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada T2N 1N4
pwlfong@ucalgary.ca

ABSTRACT

Social Network Systems pioneer a paradigm of access control that is distinct from traditional approaches to access control. Gates coined the term Relationship-Based Access Control (ReBAC) to refer to this paradigm. ReBAC is characterized by the explicit tracking of interpersonal relationships between users, and the expression of access control policies in terms of these relationships. This work explores what it takes to widen the applicability of ReBAC to application domains other than social computing. To this end, we formulate an archetypical ReBAC model to capture the essence of the paradigm, that is, authorization decisions are based on the relationship between the resource owner and the resource accessor in a social network maintained by the protection system. A novelty of the model is that it captures the contextual nature of relationships. We devise a policy language, based on modal logic, for composing access control policies that support delegation of trust. We use a case study in the domain of Electronic Health Records to demonstrate the utility of our model and its policy language. This work provides initial evidence to the feasibility and utility of ReBAC as a general-purpose paradigm of access control.

Categories and Subject Descriptors

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

General Terms

Security, Design, Language, Theory

Keywords

Contexts, electronic health records, modal logic, policy language, relationship-based access control, social networks

1. INTRODUCTION

A social network is a collection of assertions regarding the relationships between individuals in a given population. Each assertion says something about the nature of the underlying relationship. For example, an assertion "`father(john, peter)`" specifies that the father of John is Peter.

Recent years have witnessed the growing popularity of Social Network Systems (SNSs). An SNS is essentially an information sharing system that explicitly tracks the social network of its users. Where the relationship assertions come from depends on the domain of application. In some applications (e.g., Facebook), each assertion represents a consensus reached by the participants of that relationship: i.e., each assertion is jointly asserted by those participants. In other applications (e.g., Google Buzz), the assertions may be mined from user data without the consensus of those users. It is also conceivable that the assertions are maintained as part of the on-going operation of a system (e.g., professional relationships between health service providers and their clients).

An SNS maintains a social network for at least two reasons. First, it is used by the users to navigate the information space of the system (i.e., by "traversing" the "edges" of the social network). Second, the social network is used as a basis for formulating the access control policies of user-contributed resources (e.g., "this photo album is accessible only by my friends-of-friends"). It is this second use of social networks that this work focuses on.

Although SNSs have a humble beginning in social computing, we believe that they pioneer a paradigm of access control that has much wider applications. This paradigm is characterized by (1) the explicit tracking of one or more social networks by the protection system, and (2) the expression of access control policies in terms of the relationship between the resource owner and the resource accessor in these social networks. Such an access control paradigm is particularly suited for application domains in which those relationships on which authorization decisions are based arise not from the subjective assessment of the users, but from the structure of trust that is inherent in the application domain. Examples of such relationships include professional relationships as well as relationships induced by organizational structures. Following Gates [17], we call this paradigm of access control ***Relationship-Based Access Control (ReBAC)***. This work advocates the adoption of ReBAC for application-level security and privacy, and provides initial evidence for the feasibility and utility of this access control paradigm.

1.1 Motivations

Relationships as Basis of Authorization. Many traditional access control systems base their authorization decisions on unary predicates of users. For example, the accessor must have a certain identity, or a certain role. In many emerging application domains, such as healthcare and education domains, it is more natural to base authorization decisions on whether the resource owner and accessor are in a particular kind of relationship (e.g., professional relationship). This need is evident when we examine how Role-Based Access Control (RBAC) models [30, 15, 14] have been pushed to the limit to cope with this demand. For example, some previous work defines ***parameterized roles*** [29, 26] (also called ***role templates*** [19]), such as manager(john), to represent the role assumed by the manager of John. Such a parameterized role is merely a way of encoding a binary relation manager(u, v) between user u and her manager v. We believe that in some application domains, it is more natural to simply model binary relations rather than unary predicates that are further parameterized. Section 3 presents a formal access control model that features a relationship-based authorization scheme.

Composing Relations. In a standard RBAC system, when a permission p is assigned to a role R, we are essentially formulating the following policy: "*grant p to user u if $R(u)$*". Here we interpret R as a unary predicate on users (more precisely, R is induced by the user-role assignment). A natural question to ask is whether we can use combinators to compose R from simpler unary predicates. It turns out, except for boolean combinations, there is really not much one can do with unary predicates. These boolean combinations can usually be encoded by the role hierarchy with ease. Binary relations, however, are an entirely different breed of animal. In particular, binary relations are *composable*. Given binary relations R_1 and R_2, one may compose the relation $R_1 \circ R_2$. This means complex binary relations can be composed from primitive building blocks. As an example, consider the relation friends. We can derive the friends-of-friends relation by the composition friends $\circ$ friends. Relation composition is but one way of building complex binary relations from simple ones. In Section 4, we propose a policy language, based on modal logic, for supporting the specification and composition of complex binary relations.

Delegation of Trust. Composite relations are interesting because they can be employed to express delegation of trust [5, 13, 35, 26, 25, 3]. For example, by granting access to friends-of-friends, we are essentially delegating to our friends to decide who may access. Section 5 presents a case study to demonstrate this use of ReBAC policies.

Context-specific Relationships. Another consideration is the contextual nature of relationships. Many existing SNSs features a single social network as the basis of all authorization decisions. Yet one size truly does not fit all. Many relationships we encounter are contextual. A physician who is my treating physician in one medical case may very well be a consulting expert in a different medical case of mine. As a result, the physician may enjoy a different level of access in each case. This contextual nature of relationships motivates the need for a ReBAC system to track multiple access contexts. Relationships can be articulated or dissolved separately in each access context. The result is that authorization decisions may be different in each context even though the access request remains the same.

Sharing of Relationships Across Contexts. The need of tracking a distinct social network for each access context shall be balanced by the equally important need for distinct contexts to share relationships. Some relationships have a wide scope of effectiveness. The fact that Alice is a parent of Bob is significant in multiple contexts: in diagnosis for uncovering hereditary connections, in institutional registration for billing purposes, etc. There is thus a need for the access control system to support the sharing of wide-scope relationships across multiple access contexts, and to do so in a rational manner.

1.2 Contributions

This work provides initial evidence for the feasibility and utility of using ReBAC as a general-purpose protection model. Specifically, contributions of this work are the following:

1. A ReBAC model is formulated to capture the core idea of employing social networks as the basis of authorization decisions. Contrary to Facebook-style SNSs, this model tracks social networks that are poly-relational (e.g., child-parent relationships are distinct from patient-physician relationships) and directed (i.e., child-parent relationships are distinct from parent-child relationships). These features allow the model to capture rich domain concepts.

2. The model captures the context-dependence nature of relationships. Relationships are articulated in contexts, and accesses are authorized also in contexts. Sharing of relationships among contexts are achieved in a rational manner through a context hierarchy.

3. A policy language based on modal logic is proposed for expressing ReBAC policies. The language provides means for composing complex policies from simple ones.

4. A case study in the domain of Electronic Health Records (EHR) systems is conducted to demonstrate the utility of the proposed ReBAC model. The case study highlights features such as delegation of trust and scoping of relationships.

2. RELATED WORK

The term Relationship-Based Access Control and the acronym ReBAC were previously coined by Gates [17] as she articulated the protection requirements of Web 2.0 applications. According to her, "a paradigm of access control needs to be developed that is based on interpersonal relationships." Our model is one approach to meet this requirement. Other authors who coined similar terms include [22, 18, 7].

There exists a number of parallels between RBAC [30, 15, 14] and ReBAC, such as those highlighted in Section 1.1 (user-role assignment vs inter-user relationships, permission-role assignment vs policies) and Section 6 (sessions vs contexts, role hierarchy vs context hierarchy, separation-of-duty constraints vs well-formed contexts). In these two sections, we attempt to underline the fact that ReBAC is not entirely distinct from RBAC. Instead, it is a natural generalization through the use of binary relations over users rather than unary relations (i.e., roles) for capturing domain knowledge. The concrete benefits brought about by this generalization is the flexibility to compose complex policies from primitive ones, and the support of trust delegation.

Trust delegation is one of the main features of Trust Management Systems and other distributed authorization systems [5, 13, 35, 26, 25, 3]. In these systems, authorization decisions are based on declarative policy statements, the satisfaction of which is based in turn on assertions made by multiple principals. By using declarative policies and articulating user relationships, our formulation of ReBAC is highly related to trust management, allowing it to support trust delegation. Note that, however, we envision the model being used for application-level security and privacy in the same manner RBAC is used, rather than in situations necessitating distributed authorization. The model is also different from trust management in that we constrain the compatibility of assertions (i.e., relationships) through the use of a well-formed context hierarchy (see Section 6, cf. [2]), and we employ a modal logic as the policy language.

Recent years have seen the proposal of a number of access control systems or models for SNSs. We begin with the work of Kruk *et al.* [23]. D-FOAF is a distributed identity management system that employs a social network to enable trust delegation. The social network is mono-relational, with relationships that are directed and weighted (i.e., to represent the strength of relationships). Access control policies are expressed as distance and strength thresholds (e.g., allow access if owner and accessor are connected by a path of length no more than k and the aggregate strength of that path is at least δ). As the social network is essentially a weighted directed graph, authorization decisions are computed by a variant of Dijkstra algorithm. Our model tracks poly-relational social networks, and the policy language can be pushed to express distance-based policies. Support for relationship strength is rudimentary (see Section 6).

Carminati *et al.* developed a decentralized social network system that tracks poly-relational social networks in which relationships are weighted by trust levels [9, 10, 6, 11, 12]. In an early proposal [9, 11], a typical access control policy grants access when there exists a path between the owner and the accessor, consisting of relationships of a particular type, of length below a certain bound, and with aggregate trust level above a certain threshold. The final system [12] eventually adopts a distributed trust metrics, in which the trust between the owner is the accessor is obtained by the weighted average trust levels between the accessor and the "trustworthy" neighbours of the owner. In our work, access control policies are declarative and qualitative. Relationship paths may be composed of multiple types of relationships.

In [8], Carminati *et al.* proposed an access control system for social computing, in which semantic web technologies, including the Resource Description Framework (RDF) and the Web Ontology Language (OWL), are adopted to describe user profiles, relationships among users, resources, relationship between users and resources, and actions. Doing so allows them to see a social network as a knowledge base of user-user and user-resource relationships, and based on which access control policies are formulated. Our work has been partly influenced by this knowledge-based perspective, though our proposed model is agnostic to implementation and representation issues. In addition, although our model captures only user-user relationships, our work offers a declarative policy language and a context hierarchy to scope the effectiveness of relationships.

Fong *et al.* proposed an access control model that formalizes and generalizes the access control mechanism implemented in Facebook [16]. The model admits arbitrary policy vocabulary that are based on graph-theoretic properties (e.g., allow access if owner and accessor participate in the same k-clique, if owner and accessor share k common neighbours, etc). The present work differs from theirs in three ways. First, [16] models capability-like entities called search listings, the reachability of which is a necessary condition for access. In the present work, access control policies are the only condition of access. Second, the present model captures relationship types and contexts, which are not modelled in [16]. Third, in a subsequent work [1], Fong *et al.* employ birooted graphs as a policy language. Here we employ modal logic for specifying policies.

PriMa [31] is another recently proposed privacy protection mechanism for SNSs. The premise of the work is the observation that, because of the growing complexity of the social network and the proliferation of user content categories, it is perhaps wise not to rely on regular users to manually set up their access control policies. PriMa is a scheme by which access control policies are automatically constructed for users. The policy construction algorithm considers factors such as the following: average privacy preference of similar and related users, popularity of the owner (i.e., the more connected is the owner, the more sensitive its profile items), accessibility of similar items in similar and related users (i.e., if my peers do not grant me access, then I better do not grant access easily), closeness of owner and accessor (measured by the number of common friends), etc. These factors are then combined to generate access control rules for profile items. The proposed ReBAC model does not preclude the use of automatic policy inference engines such as [31]

Squicciarini *et al.* [32] considered access control policies of data that is co-owned by multiple parties in an SNS setting, such that each co-owner may separately specify her own privacy preference for the shared data. A voting algorithm was adopted to enable the collective enforcement of shared data. Game theory was applied to evaluate the scheme. In our model, we assume that there is exactly one stakeholder (i.e., owner) for each resource. Generalizing the model to cope with co-ownership is an interesting future direction.

3. A REBAC MODEL

A thesis of this work is that an SNS can be adopted as the access control system of an information sharing platform. We describe here an archetypical ReBAC model, which offers two main features.

First, the social networks tracked by this model are poly-relational, in the sense that the model tracks not only *whether* a relationship exists beween two individuals, but also the *type* of that relationship (e.g., patient-physician, parent-child, etc). This is a generalization of mono-relational networks, such as those found in Facebook-style SNSs. This generalization can be exploited to support the rich relational concepts found in many application domains. The model further captures the idea that an authorization decision is based solely on the relationship between the resource owner and the resource accessor in a certain social network.

Second, the model tracks multiple access contexts. Relationships may be articulated in separate contexts. To facilitate the sharing of relationship across contexts, the access contexts are organized into a tree-shaped hierarchy (see Figure 1). The hierarchy facilitates a sharing mechanism known as relationship inheritance. When an access is requested in an access context, the relationships articulated in all the ancestor contexts are combined with the relationships in the target access context to form a single social network. This social network is the one on which authorization decisions are made. Lastly, the creation and destruction of access contexts follow a stack discipline: new contexts are introduced as leaves, and only leaves can be removed from the tree. This features supports the scoping of the effectiveness of relationships.

An access control system is traditionally modelled as a state transition system [21, 27]. State transitions capture the management aspects of the system, including various forms of system reconfigurations. Actual accesses are performed with respect to a given state of the system [27]. In this section, we describe the global parameters to an ReBAC protection system, delineate its state space as well as state transitions, and specify the authorization procedure.

3.1 Notations

Consider a function $f : X \to Y$ and individuals x_0 and y_0, where $y_0 \in Y$, but x_0 may or may not be in X. We write $f[x_0 \mapsto y_0]$ to denote the function $f' : X \cup \{x_0\} \to Y$ defined as follows: $f'(x) = y_0$ if $x = x_0$, but $f'(x) = f(x)$ if $x \neq x_0$. Suppose further $X' \subseteq X$. We write $f \mid X'$ for the restriction of f to the domain X'. Given a binary relation $R \subseteq X \times X$ and individuals $x, y \in X$, we write $x\,R\,y$ iff $(x, y) \in R$. We also write R^* to denote the reflexive transitive closure of R.

3.2 Social Networks and Relation Identifiers

We assume that an SNS defines a countable set $\mathcal{I}$ of ***relation identifiers***. Each identifier denotes a type of relationships that is tracked by the system (e.g., parent-child, patient-physician, etc). A typical member of $\mathcal{I}$ is denoted by i.

A social network is essentially a directed graph with multiple kinds of edges. While individuals are represented by vertices, each kind of directed edges represents a distinct type of relationship between users. Formally, a ***social network*** G is a relational structure $\langle V, \{R_i\}_{i\in\mathcal{I}}\rangle$, where [4]:

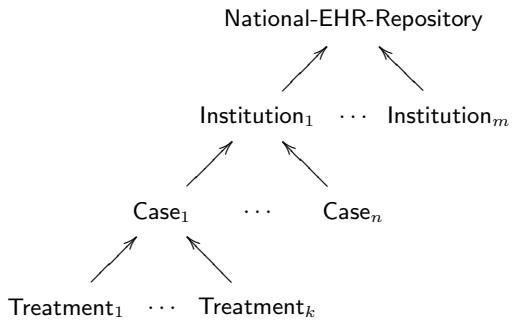

Figure 1: A sample hierarchy of access contexts in an EHR application.

- V is a finite set of vertices, each representing an individual in the social network.
- $\{R_i\}_{i\in\mathcal{I}}$ is a family of binary relations. The binary relation $R_i \subseteq V \times V$ specifies the pairs of individuals participating in relationship type i.

In the definition above, the binary relations need not be irreflexive or symmetric: i.e., the graph may contain circles, and the relationships may not be reciprocal. This is more general than Facebook-style SNSs, in which the social network is a simple graph [16].

We denote by $\mathcal{G}(V, \mathcal{I})$ the set of all social networks defined over vertex set V and relation index set $\mathcal{I}$.

We introduce some additional conventions and notations related to relation identifiers and social networks before we move on to discuss access control policies.

We adopt the following convention to name relation identifiers. We assume that, with every binary relation, there are named roles for each of the two participants. For example, in a parent-child relationship, the parent and the child are the two roles. We will use the receiving role to name the relation. That is, the relation identifier child names the parent-child relation, while the identifier parent names the inverse relation (i.e., the child-parent relation).

A social network is ***inverse-closed*** whenever the following holds: for every $i \in \mathcal{I}$, there is an identifier $-i \in \mathcal{I}$ such that $R_{-i} = {R_i}^{-1}$. That is, the inverse of a relation is always defined in the social network. For example, if, for a given social network, $\mathcal{I} = \{\mathsf{parent}, \mathsf{child}, \mathsf{physician}, \mathsf{patient}\}$, and the identifiers denote respectively the child-parent, parent-child, patient-physician and physician-patient relations, then the social network is inverse-closed. Unless stated otherwise, we consider only inverse-closed social networks. In such a case, we omit the inverses when we enumerate $\mathcal{I}$, using the following notational convention: $\mathcal{I} = \{\mathsf{parent}, \mathsf{physician}, -\ldots\}$.

Suppose $G = \langle V, \{R_i\}_{i\in\mathcal{I}}\rangle$ is a social network, and $\Delta = \{R'_i\}_{i\in\mathcal{I}}$ is a family of binary relations defined over the vertex set V. Intuitively, Δ represents edges for a social network defined over V. We write $G+\Delta$ to denote the social network

$\langle V, \{R_i \cup R'_i\}_{i\in\mathcal{I}}\rangle$, that is, the social network obtained from G by adding the edges in Δ. Similarly, we write $G - \Delta$ to denote $\langle V, \{R_i \backslash R'_i\}_{i\in\mathcal{I}}\rangle$ (i.e., the social network obtained from G by deleting the edges in Δ).

Given two social networks $G = \langle V, \{R_i\}_{i\in\mathcal{I}}\rangle$ and $G' = \langle V', \{R'_i\}_{i\in\mathcal{I}}\rangle$, we write $G \cup G'$ to denote the social network $\langle V \cup V', \{R_i \cup R'_i\}_{i\in\mathcal{I}}\rangle$. This social network, called the ***union*** of G and G', is obtained by superimposing G and G' on one another. The binary operation $\cup$ is obviously commutative and associative. Suppose $\mathcal{G} = \{G_1, \ldots, G_k\}$ is a finite set of social networks. We write $\bigcup \mathcal{G}$ to denote $G_1 \cup \ldots \cup G_k$.

3.3 Access Control Policies

An SNS controls accesses initiated by ***users***. Let $\mathcal{U}$ be the set of all ***user identifiers*** (or simply ***users***) in the system. We denote typical members of $\mathcal{U}$ by u and v. Accesses are directed against ***resource identifiers*** (or simply ***resources***). A resource may represent one or more objects or certain system operations[1]. Let $\mathcal{R}$ be the set of resources protected by the SNS. A typical member of $\mathcal{R}$ is denoted by r.

Associated with every access request are therefore the following: (a) a protected resource $r \in \mathcal{R}$ that is being accessed, (b) the ***owner*** $u \in \mathcal{U}$ of that resource, and (c) the ***accessor*** $v \in \mathcal{U}$ who requests the access. Note that we use the term "owner" in a sense different from how it is used in Discretionary Access Control (DAC) [20, 27]. In DAC, the owner of an object may explicitly grant access of the object to other users in the system. This sense of controlling to whom access shall be granted is not the main focus of our usage (although it is entire possible that the system allows the owner of a resource to specify the policy of access at her own discretion, as we shall point out in Section 3.4). On the contrary, when we say that a user is the owner of a resource, we mean that an accessor must be in a specific kind of relationship with the owner (with respect to a certain social network) in order to be granted access. The marking of an individual as the anchoring end of such a relationship-based authorization procedure is the primary sense of our usage of the term "owner".

Associated with every resource is an ***access control policy***. Such a policy is modelled as a ternary predicate of the following signature: $\mathcal{U} \times \mathcal{U} \times \mathcal{G}(\mathcal{U}, \mathcal{I}) \to \{0, 1\}$. Specifically, given an owner, an accessor, and a social network, the predicate returns a boolean value to indicate whether access shall be granted. We write $\mathcal{PP}(\mathcal{U}, \mathcal{I})$ to denote the set of all policy predicates with the above signature.

3.4 Protection System

A ***protection system*** (or simply a ***system***) N is a 7-tuple $\langle \mathcal{I}, \mathcal{U}, \mathcal{R}, \mathcal{C}, c_0, \mathit{policy}, \mathit{owner}\rangle$, where:

- $\mathcal{I}$ is the set of relation identifiers, as discussed above.
- $\mathcal{U}$ is a finite set of users in the system.
- $\mathcal{R}$ is a finite set of resources to be protected by the system. We assume that the universe of user and resource identifiers remain constant throughout system lifetime (e.g., with fixed bit length), as creation and destruction of users and resources are not the focus of this modelling exercise. Note that this does not pose a real restriction to the system, as user creation can be readily modelled by a dormant user turning active.
- $\mathcal{C}$ is a *countably infinite* universe of ***access contexts*** (or simply ***contexts***). An access is requested in the backdrop of a specific access context. Each context may give a different authorization decision even if the owner, accessor and resource involved in the access are the same. As we shall see in the sequel, contexts can be created or destroyed during system execution.
- The ***root context*** $c_0 \in \mathcal{C}$ is a distinguished context. When we introduce the context hierarchy below, we shall see that c_0 is the root of the context hierarchy.
- The function $\mathit{policy} : \mathcal{R} \to \mathcal{PP}(\mathcal{U}, \mathcal{I})$ assigns a policy predicate to every resource in the system. As policy revision is not a focus of the present modelling exercise, we assume that policy settings remain constant throughout system lifetime, and model the settings as a static parameter of the system. For an example of how policy revision could have been incorporated into the modelling of an SNS, see [16].
- There is a function $\mathit{owner} : \mathcal{R} \to \mathcal{U}$ that assigns an owner to every resource in the system.

In the definition above, we do not speculate on where the access control policy of a resource comes from. There are at least three possibilities:

Mandatory. Some of the resources may have policies mandated by the system administrator.

Discretionary. For other resources, the resource owners are responsible for specifying their access control policies.

Policy Vocabulary. A moderate position, which is adopted by many existing SNSs, is for the system to mandate a ***policy vocabulary***, that is, a set of "canned" policy predicates (e.g., friends, friends-of-friends), from which users take their picks.

3.5 Protection State

Given a protection system $N = \langle \mathcal{I}, \mathcal{U}, \mathcal{R}, \mathcal{C}, c_0, \mathit{policy}, \mathit{owner}\rangle$, a ***protection state*** (or simply a ***state***) γ is a triple $\langle C, \mathit{sn}, \mathit{extends}\rangle$ composed of the following elements:

- $C \subseteq \mathcal{C}$ is the set of ***active contexts*** in the state. It is required that this set is finite and non-empty, containing at least the element c_0. Each active contexts define a scope of effectiveness for some articulated relationships.
- There is a function $\mathit{sn} : C \to \mathcal{G}(\mathcal{U}, \mathcal{I})$ that maps each context of the state to a social network. Specifically, the social network $\mathit{sn}(c)$ records the relationships that

[1] In standard RBAC literature, what we call a resource is called a permission [30, 15, 14]. We avoid the term "permission" because "permission owner" is not natural in English.

have been articulated in context c. By articulating a relationship in a context, one is placing the relationship in the scope of effectiveness represented by that context.

- $extends \subseteq C \times C$ is a binary relation defined over $\mathcal{C}$, such that (a) the directed graph $(C, extends)$ is a tree, (b) c_0 is the root of the tree, and (c) if $(c_1, c_2) \in extends$ then c_1 is the child of c_2 in the tree. States satisfying these three conditions are said to be ***well-formed***. The *extends* relation defines a ***context hierarchy***. A tree-shaped context hierarchy corresponds to the nested structure of relationship scopes. Relationships articulated in c will be visible to the authorization decisions made in c and all the descendent contexts of c.

  Using the notational conventions in Section 3.1, we also write "$c_1\ extends\ c_2$" iff c_1 is a child of c_2, and "$c_1\ extends^*\ c_2$" iff c_1 is either c_2 or one of the descendants of c_2.

We denote the set of all well-formed states for N by $\mathcal{ST}(N)$.

3.6 Authorization

Authorization is achieved by consulting relationships in a social network. In the following, we will specify the social network on which authorization decisions are based.

Given a state $\gamma = \langle C, sn, extends\rangle$, the ***effective social network*** of an access context $c \in C$ is defined as the following social network:

$$esn_\gamma(c) = \bigcup\{sn(c') \mid c' \in C,\ c\ extends^*\ c'\}$$

Specifically, $esn_\gamma(c)$ is obtained by taking the union of all social networks $sn(c')$, where c' is either the context c or one of the ancestors of c in the context hierarchy. In essence, G contains not only the relationships of $sn(c)$, but also the relationships of the social networks associated with the ancestor access contexts. The effective social network of a context is the social network that will be used as the basis of authorization decisions made in that context.

Accesses are modelled as queries over states. Specifically, a query has the following syntax:

$$q ::= v\ \textbf{accesses}\ r\ \textbf{in}\ c$$

Let $\mathcal{Q}(N)$ be the set of all queries for system N.

The sequent "$\gamma \vdash_N q$" asserts the successful authorization of access q in state γ of system N [2]:

$$\frac{\begin{array}{ccc} N = \langle \_,\_,\_,\_,\_, policy, owner\rangle & & \gamma = \langle C, \_, \_\rangle \\ & c \in C & \\ P = policy(r) & u = owner(r) & G = esn_\gamma(c) \\ & P(u, v, G) & \end{array}}{\gamma \vdash_N v\ \textbf{accesses}\ r\ \textbf{in}\ c}\ (\textsc{Auth})$$

[2]We adopt a Prolog-style convention and write an underscore "_" in place of a variable whenever that variable is not significant.

When accessor v requests to access resource r in access context c, the system will look up the policy predicate $P = policy(r)$ associated with the resource, as well as the resource owner $u = owner(r)$. The effective social network G for context c is then derived. The system applies the predicate P to the owner u, the accessor v, and the social network G in order to arrive at an authorization decision.

The rule above captures two important ideas of this model. First, it captures the idea that authorization decisions are made primarily by consulting the relationship between the accessor and the owner. In order words, only the social network aspect of the system state is significant in authorization decisions. In a real implementations, relationships may not be the sole basis for authorization decisions. It is entirely possible for the system to have a hybrid authorization scheme that is both relationship based and, for instance, role based. In this work we consider a pure form of ReBAC.

Second, the rule above also captures the notion of relationship inheritance, allowing relationships articulated in ancestor contexts to be inherited by the effective social network of a descendent context. An important consequence of this particular design is the monotonic nature of relationship inheritance. Suppose $c\ extends\ c'$. Let social networks $G = \langle \mathcal{U}, \{R_i\}_{i\in\mathcal{I}}\rangle$ and $G' = \langle \mathcal{U}, \{R'_i\}_{i\in\mathcal{I}}\rangle$ be $esn_\gamma(c)$ and $esn_\gamma(c')$ respectively. Then $R'_i \subseteq R_i$ for all $i \in \mathcal{I}$. That is, the effective social network of a child context contains no less relationships than that of a parent context.

3.7 State Transition

State transitions capture the evolution of the context hierarchy, as well as the mutation of social networks. Intuitively, the creation and destruction of access contexts follow a stack discipline, thereby allowing users to model the nested scopes of social relationships.

We proceed to define a state transition relation to specify the dynamic behaviour of a system. In fact, what we define is an *upper bound* of the actual transition relation, in the following sense. An instantiation of this access control model will *refine* the transition relation specified below, in such a way that (a) certain transitions described below may not be allowed, but (b) the refined relation will never allow transitions that are not allowed by the following specification. The additional restrictions imposed by an instantiation are usually motivated by the access control requirements of a specific application domain. We will return to this matter in the sequel.

Given a system $N = \langle \mathcal{I}, \mathcal{U}, \mathcal{R}, \mathcal{C}, c_0, policy, owner\rangle$, we define the transition relation $\cdot \xrightarrow{\cdot}_N \cdot \subseteq \mathcal{ST}(N) \times \mathcal{T}(N) \times \mathcal{ST}(N)$, where $\mathcal{T}(N)$ is a set of ***transition identifiers*** with the following syntax:

$$t ::= \mathsf{push}(c, c, \Delta) \mid \mathsf{pop}(c) \mid \mathsf{edge}(c, \Delta, \Delta)$$

where c is a context and Δ is a family of binary relations.

The transition relation is specified via three rules, the first of which models the creation of contexts:

$$\frac{\begin{array}{c} c_1 \in \mathcal{C}\backslash C \qquad c_2 \in C \\ C' = C \cup \{c_1\} \\ sn' = sn[c_1 \mapsto \langle \mathcal{U}, \Delta \rangle] \\ extends' = extends \cup \{(c_1, c_2)\} \end{array}}{\langle C, sn, extends \rangle \xrightarrow{\mathsf{push}(c_1, c_2, \Delta)}_N \langle C', sn', extends' \rangle} \text{(PUSH)}$$

The PUSH rule specifies the semantics of the operation $\mathsf{push}(c_1, c_2, \Delta)$, which adds a new leaf c_1 to the context hierarchy. The new context is a child of the existing context c_2. We are also given the option of initializing the new context with relationships in the relation family Δ.

The POP rule below models the destruction of contexts.

$$\frac{\begin{array}{c} c \in C\backslash\{c_0\} \qquad \neg\, \exists c' \in C \,.\, c' \; extends \; c \\ C' = C\backslash\{c\} \\ sn' = sn \,|\, C' \\ extends' = extends \cap (C' \times C') \end{array}}{\langle C, sn, extends \rangle \xrightarrow{\mathsf{pop}(c)}_N \langle C', sn', extends' \rangle} \text{(POP)}$$

The operation $\mathsf{pop}(c)$ removes a leaf context c from the context hierarchy. Relationships articulated in a context are dissolved when the context is removed, marking the end of the scope of those relationships. Note that the root context c_0 cannot be removed, nor can one remove a context that is not a leaf. The latter restriction ensures a last-in-first-out discipline.

The EDGE rule models the revision of social networks.

$$\frac{sn' = sn[c \mapsto (sn(c) + \Delta_1) - \Delta_2]}{\langle C, sn, extends \rangle \xrightarrow{\mathsf{edge}(c, \Delta_1, \Delta_2)}_N \langle C, sn, extends' \rangle} \text{(EDGE)}$$

The operation $\mathsf{edge}(c, \Delta_1, \Delta_2)$ adds relation family Δ_1 to, and removes relation family Δ_2 from, the social network associated with the context c.

As we pointed out, an instantiation of this model shall define a transition relation that is a refinement of the one specified above. Additional restrictions may arise from the following needs. First, there may be domain-specific constraints to the shape of the context hierarchy. For example, the hierarchy in Figure 1 is constrained to have a maximum height of four. These constraints cause certain transitions allowed by the PUSH and POP rules to become illegitimate. Second, there may be restrictions on what relationships may be articulated in each context. In the example of Figure 1, supervisory relationships shall only be articulated in an Institution context. Such restrictions cause certain transitions allowed by the EDGE and PUSH rules to become illegitimate.

We do not explicitly track the initiator of each transition, because the authorization of transitions is not a focus of this work. In principle, the transitions can be considered resources and thus protected by the same protection system.

Lastly, the transition relation defined above (or any of its refinements) preserves the three well-formedness conditions specified in Section 3.5.

4. A MODAL APPROACH TO REBAC POLICY SPECIFICATION

Section 3.4 outlines a number ways by which access control policies originate in a ReBAC system. In many situations, it is desirable to have a policy language for specifying ReBAC policies. First, a policy language facilitates the specification of composite policies, which in turn forms the basis of trust delegation. Second, a policy language facilitates the static analysis of policies and system configuration, an agenda we plan to pursue as future work. Our goal in this section is to devise a policy language for expressing ReBAC policies, and we propose to adopt a modal logic for this purpose.

A ReBAC policy predicate describes the relationship between an owner and an accessor in a social network, which in turn can be naturally captured by a relational structure. A modal logic is essentially a language for describing the topological properties of a relational structure. Specifically, a modal logic provides "an internal, local perspective on relational structures" [4]. That is, a modal formula specifies topological properties from the perspective of a specific vertex in a relational structure, by describing how the relational structure appears to a "crawler" locating at that vertex. Such a perspective is particularly useful for the specification of ReBAC policies: a modal formula could be employed to specify how an owner relates to potential accessors in a social network. We back up this claim by introducing a basic modal language for specifying ReBAC policies.

4.1 Syntax and Semantics

A formula in our basic modal language expresses a desired relationship between an owner and an accessor in a given social network. The syntax of the language is given below.

$$\phi, \psi ::= \top \mid \mathsf{a} \mid \neg\phi \mid \phi \vee \psi \mid \langle i \rangle \, \phi$$

where $i \in \mathcal{I}$ is a relation identifier. The formula $\top$ is the constant true, and is satisfied by any pair of owner and accessor. The atomic formula a asserts that the accessor is the owner herself. The connectives $\neg$ and $\vee$ are the usual boolean negation and disjunction. For example, $\neg\phi$ asserts that the owner and the accessor do not participate in the relationship specified by ϕ. The formula $\langle i \rangle \, \phi$ asserts that the owner is related to a vertex via an i relationship, and that vertex is in turn related to the accessor in a manner specified by the formula ϕ.

The formal semantics of formulas are captured by the satisfaction relation $(G, u, v \models \phi)$, which asserts that, in social network $G = \langle V, \{R_i\}_{i \in \mathcal{I}} \rangle$, owner $u \in V$ and accessor $v \in V$ are related in a manner specified by formula ϕ:

- $G, u, v \models \top$
- $G, u, v \models \mathsf{a}$ iff $u = v$.
- $G, u, v \models \neg\phi$ iff it is not the case that $G, u, v \models \phi$
- $G, u, v \models \phi \vee \psi$ iff either $G, u, v \models \phi$ or $G, u, v \models \psi$
- $G, u, v \models \langle i \rangle \, \phi$ iff there exists $u' \in V$ such that $(u, u') \in R_i$ and $G, u', v \models \phi$

We also introduce derived forms to capture the duals of the above constructs.

$$\bot = \neg\top \quad \bar{\mathsf{a}} = \neg\mathsf{a} \quad \phi \wedge \psi = \neg(\neg\phi \vee \neg\psi) \quad [i]\,\phi = \neg\,\langle i\rangle\,\neg\phi$$

Intuitively, $\bar{\mathsf{a}}$ means that the accessor is not the owner, and $[i]\,\phi$ means that every individual related to the owner via an i relationship is also related to the accessor in a manner specified by ϕ.

The language above deviates from a typical basic modal language in two ways. Firstly, there are only two propositional symbols, namely, a and $\bar{\mathsf{a}}$. Other basic modal languages generally support multiple propositional symbols. Secondly, our satisfaction relation relates two individuals in a relational structure. On the contrary, the satisfaction relation of a standard modal language asserts a property of one individual in the context of a relational structure.

We write $[\![\phi]\!]$ to denote the predicate $P(u, v, G)$ that returns true iff $G, u, v \models \phi$. One can now use modal formulas to specify policy predicates, for example, by setting $policy(r) := [\![\phi]\!]$.

4.2 Examples

We illustrate how our modal language can be used for specifying ReBAC policies. Consider an SNS with relation identifiers $\mathcal{I} = \{\mathsf{parent}, \mathsf{sibling}, \mathsf{spouse}, -\ldots\}$ and inverse-closed social networks. Suppose further we are to specify a ReBAC policy $[\![\phi]\!]$ for a resource r. In the following, we will state a number of policies first in English, and then provide a choice of ϕ that captures the English specification.

"*Grant access to the owner's spouse.*" The policy can be expressed by the formula "$\langle\mathsf{spouse}\rangle\,\mathsf{a}$". Essentially, the idiom "$\langle i\rangle\,\mathsf{a}$" asserts that successful accessors must be related to the owner directly via an i relationship.

"*Graph access to the owner's child.*" The policy can be expressed by the formula "$\langle\mathsf{\text{-}parent}\rangle\,\mathsf{a}$". This example demonstrates how one can name the inverse of a relation in an inverse-closed environment.

"*Grant access to grand parents.*" The formula that expresses this policy is "$\langle\mathsf{parent}\rangle\,\langle\mathsf{parent}\rangle\,\mathsf{a}$". This example illustrates how composite relationships can be expressed.

"*Grant access to parents, aunts and uncles.*" A possible formula is:

$$\langle\mathsf{parent}\rangle\,\mathsf{a} \vee \langle\mathsf{parent}\rangle\,\langle\mathsf{sibling}\rangle\,\mathsf{a} \vee \langle\mathsf{parent}\rangle\,\langle\mathsf{sibling}\rangle\,\langle\mathsf{spouse}\rangle\,\mathsf{a}$$

This example illustrates the use of boolean connectives.

"*Grant access unless the accessor is a parent of the owner.*" A possible formula is:

$$\neg\,\langle\mathsf{parent}\rangle\,\mathsf{a}$$

Another possible formula is:

$$[\mathsf{parent}]\,\bar{\mathsf{a}}$$

This example illustrates the duality of $[i]$ and $\langle i\rangle$.

"*Grant access to a sibling who is not married.*" A formula to express the policy is:

$$\langle\mathsf{sibling}\rangle(\mathsf{a} \wedge [\mathsf{spouse}]\,\bot)$$

Note that the idiomatic phrase "$[\mathsf{spouse}]\,\bot$" asserts that the matching sibling "has no spouse".

"*Grant access to a married sibling.*" A formula to express this policy is:

$$\langle\mathsf{sibling}\rangle(\mathsf{a} \wedge \langle\mathsf{spouse}\rangle\,\top)$$

The idiomatic phrase "$\langle\mathsf{spouse}\rangle\,\top$" asserts that the matching sibling "has a spouse".

"*Grant access if accessor is the only child of the owner.*" A formula to express the policy is:

$$\langle\mathsf{\text{-}parent}\rangle\,\mathsf{a} \wedge [\mathsf{\text{-}parent}]\,\mathsf{a}$$

4.3 Model Checking

Suppose $policy(r) = [\![\phi]\!]$. Then the authorization of the request "v **accesses** r **in** c" in state γ involves testing $G, u, v \models \phi$, where $G = esn_\gamma(c)$ and $u = owner(r)$. This test is known in the literature as local model checking [33]. The definition of the satisfaction relation can be interpreted procedurally as a recursive algorithm for model checking. Whenever a modal operator "$\langle i\rangle$" or "$[i]$" is encountered in the process, a query of the form "$(u, ?x) \in R_i$" will be directed against the social network $esn_\gamma(c)$, where "$?x$" is an uninstantiated variable. The result of the query is a (possibly empty) list of compatible bindings for $?x$. The time complexity of the algorithm, measured in terms of the number of queries made, is essentially that of depth-first search, the search tree of which has a height bounded by the maximum level of nesting of modal operators in ϕ. Due to the small world phenomenon [34], we believe the nesting of modal operators will be moderate.

To apply the above recursive procedure, one needs to have access to $esn_\gamma(c)$. Fortunately, it is not necessary for the system to explicitly construct $esn_\gamma(c)$. Specifically, a query "$(u, ?x) \in R_i$" against $esn_\gamma(c)$ can be compiled into queries against social networks $sn(c')$, for all ancestor contexts c' for which $c\ extends^*\ c'$. Because well-formed context hierarchies are trees, the number of relevant contexts is linear to the height h of the context hierarchy. This adds a factor of $O(h)$ to the time complexity of model checking. In our running example, as we also anticipate to be the case in many application domains, h is bounded by a constant.

5. A CASE STUDY: ELECTRONIC HEALTH RECORDS

To demonstrate the utility of the proposed model, this section presents a case study in the domain of Electronic Health Records (EHR) systems. Specifically, we adapt the EHR case study originally proposed in [3] to illustrate the use of the proposed model[3].

Consider an EHR system owned by some National Health Authority. The proposed ReBAC model is instantiated for

[3]Two elements of the original study have been left out in our adaptation: (a) handling of exceptions and "break-the-seal" policies, and (b) policies regulating the disclosure of third-party contributed information regarding a patient.

this EHR system. Specifically, there are four kinds of contexts in our instantiation (see Figure 1). At the root of the context hierarchy is the National-EHR-Repository context, which tracks relationships of high degree of permanence, and of global significance. The children of the root are Institution contexts. They track relationships specific to clinical institutions. An Institution context is created when that institution comes into existence. The children of an Institution context are Case contexts, one for each medical case. The relationships tracked in such a context describe responsibilities of clinicians attending to that case. Lastly, each Case context may have children that are Treatment contexts. These contexts track relationships that are specific to the administration of a particular treatment. The hierarchy has a maximum depth of four.

5.1 Treating Clinicians

Suppose the National Health Authority is to mandate a default access control policy for patient records. That is, for each patient record r, we are to come up with a reasonable policy formula ϕ so that we can set $policy(r)$ to $[\![\phi]\!]$.

The crux of the problem is to characterize the relation between a patient and her ***treating clinicians***. The treating clinicians of a patient are those clinicians responsible for treating the patient. They should be granted the right to examine the patient's records. Our goal is therefore to produce a formula *treating-clinician* that captures this relation. We build this formula incrementally.

General Practitioner. *Bob visits Zoe, his General Practitioner (GP), about his heart problem. Zoe needs to be able to access Bob's records.*

The first candidate of a treating clinician is the GP of a patient. When Zoe first became the GP of Bob, she requested Bob to give his consent to treatment. This was achieved by adding the pair $(\mathsf{Bob}, \mathsf{Zoe})$ to the binary relation R_{gp} of the social network $sn(\mathsf{National\text{-}EHR\text{-}Repository})$. In essence, an edge of type gp is added to the social network in the National-EHR-Repository context to link Bob to Zoe. This can be achieved by an EDGE transition.

Given this set-up, the formula *treating-clinician* can be defined as follows:

$$\textit{treating-clinician} = \langle \mathsf{gp} \rangle\, \mathsf{a}$$

The formula identifies the GP to be a treating clinician of a patient.

Referral. *Zoe refers Bob to a local hospital's cardiologist, Hannah. This referral shall enable Hannah to access Bob's records. Bob's express consent shall not be needed.*

Hannah creates a new case for Bob's heart problem. This is achieved by using the PUSH rule to create a new Case context as a child of the hospital's Institution context. The intention is that all relationships applicable only to this case are articulated in this context. When the case closes, all such relationships will dissolve.

Zoe and Hannah then jointly articulate the referral relationship by adding a referrer type edge $(\mathsf{Hannah}, \mathsf{Zoe})$ to the social network in the Case context of Bob's heart problem.

With this set-up in mind, the formula *treating-clinician* will be revised as follows:

$$\textit{treating-clinician} = \langle \mathsf{gp} \rangle\, \mathsf{a} \vee \langle \mathsf{gp} \rangle \langle \text{-}\mathsf{referrer} \rangle\, \mathsf{a}$$

When Hannah attempts to access Bob's records in the appropriate Case context, she will succeed. The express consent of Bob is not required. Delegation of trust occurs through the joint consent of Zoe and Hannah. Also, this delegation will be revoked when the heart problem case closes.

Surgical Team. *Hannah prescribes a heart bypass operation. A surgical team is assembled, with Lily as the lead of the team, which is further composed of other clinicians. The team needs access to Bob's records.*

Hannah uses the PUSH rule to a create Treatment context for the heart bypass operation. This Treatment context is a child of the Case context corresponding to Bob's heart problem. An appoint-team type edge $(\mathsf{Hannah}, \mathsf{Lily})$ will be added to the social network of this Treatment context. Also, a member type edge (Lily, v) will be added to the Treatment social network for each member v of the surgical team. In essence, Lily acts as a representative of the surgical team.

The above set-up leads to the following revision of the *treating-clinician* formula:

$$\begin{aligned}\textit{treating-clinician} = {} & \langle \mathsf{gp} \rangle\, \mathsf{a} \vee \langle \mathsf{gp} \rangle \langle \text{-}\mathsf{referrer} \rangle\, \mathsf{a} \\ & \vee \langle \mathsf{gp} \rangle \langle \text{-}\mathsf{referrer} \rangle \langle \mathsf{appoint\text{-}team} \rangle (\mathsf{a} \vee \langle \mathsf{member} \rangle\, \mathsf{a})\end{aligned}$$

Such a policy gives access to the entire surgical team, including both Lily and other team members. Note that this access will be revoked when the Treatment context is destroyed.

Ward Nurses. *While Bob is recovering in his ward, the ward nurses need access to his records. Bob's ward is under the supervision of the head nurse Nancy.*

In the Institution context, the hospital articulates a ward-nurse type edge (Nancy, v) for each nurse v under the supervision of Nancy. When Bob first registered into the hospital, a register-ward edge $(\mathsf{Bob}, \mathsf{Nancy})$ is articulated in the Institution context.

The set-up above allows us to add another disjunct to the *treating-clinician* formula (for brevity we do not repeat the other disjuncts):

$$\begin{aligned}\textit{treating-clinician} = {} & \ldots \vee \\ & \langle \mathsf{register\text{-}ward} \rangle (\mathsf{a} \vee \langle \mathsf{ward\text{-}nurse} \rangle\, \mathsf{a})\end{aligned}$$

The policy grants access to Nancy and the nurses in her ward so long as Bob stays in that ward.

5.2 Agent

Another resource to be protected is agency. Specifically, a patient needs to declare who will act for her when she is incapacitated. (Note that this policy is formulated under the discretion of the patient, rather than mandated by the system.) We want to formulate a policy formula *agent* that captures the relation between a patient and her agent.

During Bob's operation, complications arise. Bob needs to be kept in artificial coma. Zoe appoints Bob's wife, Carol, to be his agent.

Prior to the incident, Bob formulated the following policy for agency:

$$\mathsf{a} \vee \langle \mathsf{agent} \rangle \, \mathsf{a}$$

The policy allows either Bob or his agent to act for him.

When Bob is in comma, Zoe adds an edge $(\mathsf{Bob}, \mathsf{Carol})$ of type agent into the root context National-EHR-Repository, thereby allowing Carol to act for Bob. Because of the global nature of the root context, the agent edge needs to be explicitly removed when Bob recovers and deems the agency relationship no longer applicable.

6. DISCUSSIONS

Throughout this work we have assumed that there exists a standard taxonomy of relationship types specific to the domain of application. The taxonomy is represented by the set $\mathcal{I}$. This assumption is shared by previous work [12, 8]. We believe this assumption is reasonable, as the taxonomy can be developed as part of the requirement engineering process, just like the role hierarchy requires careful engineering.

Relationship identifiers can provide qualitative representation of relationship strength: e.g., $\mathcal{I} = \{\mathsf{acquaintance}, \mathsf{friend}, \mathsf{bff}\}$. With some clumsiness, some policies that are based on relationship strength can be expressed in our policy language: e.g., $\langle \mathsf{friend} \rangle \, \langle \mathsf{bff} \rangle \, \mathsf{a}$. In application domains in which accessibility is based on professional relationships, we do not anticipate there is a need for strength-based policies.

Contexts and context inheritance could be used for modelling various domain concepts, including the following: (1) administrative units and organizational structures, such as institutions, departments, teams, etc; (2) task-like entities, such as cases, treatments, transactions, etc; (3) episodic entities such as stages, periods, etc.

In RBAC, every access is performed in a session, in which roles are activated. An analogous arrangement in our model is that every access must be performed in a context. Doing so "activates" the relationships that are articulated in either that context or one of the ancestor contexts. The activated relationships are then used for authorizing the requested access. Thus, contexts capture sets of relationships that can be legitimately "activated" at the same time. The requirement that the context hierarchy must be well-formed (i.e., in tree shape) specifies the compatibility of relationships: relationships residing in different branches of the hierarchy cannot be "activated" at once. This is analogous to RBAC constraints over what roles can be activated simultaneously in the same session (e.g., separation of duty [28]).

7. LIMITATIONS AND FUTURE WORK

The context hierarchy assumes a tree shape: i.e., only single inheritance is permitted. Multiple inheritance corresponds to a more flexible means of constraining when relationships can be "activated" simultaneously. In principle, multiple inheritance can incorporated into the model with ease. The only problem is that model checking could become intractable, as the number of social networks that must be consulted to evaluate a policy formula could be exponential to the height of the hierarchy. An research problem is to identify a controlled form of multiple inheritance that could lead to efficient model checking.

This work relies on the user to select an appropriate context to initiate access. Context identification mechanisms can be designed to suggest appropriate contexts to users. Such a mechanism may involve the translation of location, time, proximity, device and network information into one or more plausible contexts [24]. Alternatively, one may embed ReBAC in a Workflow Management System, and map workflow tasks to contexts. We leave the detailed design of these mappings to future work.

An outstanding issue is whether the modal language can express all the ReBAC policies one desires to express. This is the problem of ***representational completeness***. Modal languages capture the idea of bisimulation [4], but [1] found that some useful relational policies are characterized by graph isomorphism instead of bisimulation. Can the modal language be extended to make it representationally complete? Besides modal languages, what are other linguistic devices that can be exploited to facilitate relation composition?

In this work we focus on binary relations because they can be readily composed. It is conceivable that our model can be generalized to incorporate relations of higher-arity. A question is whether doing so really brings additional benefits in terms of expressiveness (as every relation can be encoded by one or more binary relations). Another challenge is to fashion appropriate modal operators in the policy language when relations of arbitrary arity are involved (in principle this can be done [4]).

It is conceivable that contexts may correspond to geographically separated administrative domains. This means model checking must be conducted by consulting social networks that are stored in a distributed manner. How do we design efficient model checking algorithms when distributed storage is taken into account?

8. SUMMARY

This work advocates the use of Relationship-Based Access Control in application domains in which binary relations are more natural for expressing authorization decisions than unary relations (e.g., roles). To demonstrate the feasibility of this approach, we proposed an access control model that bases authorization decisions on the relationships between the resource owner and the resource accessor in a social network. The model features the notion of access contexts for capturing the contextual nature of relationships, and employs a context hierarchy to facilitate the rational sharing of relationships between contexts. A modal language have been proposed to facilitate the specification and composi-

tion of ReBAC policies. A case study in the domain of EHR systems has been presented to demonstrate the utility of this approach. We have thus provided initial evidence on the feasibility and utility of ReBAC as a general-purpose protection approach for application security and privacy.

Acknowledgements

This work is supported in part by an NSERC Strategic Project grant. The author would like to thank Jason Crampton for some early discussions, and Ida Siahaan for her careful reading of a draft of this work.

9. REFERENCES

[1] Mohd Anwar, Zhen Zhao, and Philip W. L. Fong. An access control model for Facebook-style social network systems. Technical Report 2010-959-08, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, July 2010. Submitted for review.

[2] Lujo Bauer, Limin Jia, and Divya Sharma. Constraining credential usage in logic-based access control. In *Proceedings of the 23rd IEEE Computer Security Foundations Symposium (CSF'10)*, pages 154–168, Edinburgh, UK, July 2010.

[3] Moritz Y. Becker and Peter Sewell. Cassandra: Flexible trust management, applied to electronic health records. In *Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW'04)*, Pacific Grove, California, USA, June 2004.

[4] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge, 2001.

[5] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy (S&P'96)*, pages 164–173, Oakland, California, USA, May 1996.

[6] Barbara Carminati and Elena Ferrari. Privacy-aware collaborative access control in web-based social networks. In *Proceedings of the 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DAS'08)*, volume 5094 of *LNCS*, pages 81–96, London, UK, July 2008. Springer.

[7] Barbara Carminati and Elena Ferrari. Enforcing relationships privacy through collaborative access control in web-based social networks. In *Proceedings of the 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'09)*, Washington DC, USA, November 2009.

[8] Barbara Carminati, Elena Ferrari, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thurainsingham. A semantic web based framework for social network access control. In *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies (SACMAT'09)*, pages 177–186, Stresa, Italy, June 2009.

[9] Barbara Carminati, Elena Ferrari, and Andrea Perego. Rule-based access control for social networks. In *Proceedings of the OTM 2006 Workshops*, volume 4278 of *LNCS*, pages 1734–1744, October 2006.

[10] Barbara Carminati, Elena Ferrari, and Andrea Perego. Private relationships in social networks. In *Proceedings of Workshops in Conjunction with the International Conference on Data Engineering – ICDE'07*, pages 163–171, Istanbul, Turkey, April 2007. Springer.

[11] Barbara Carminati, Elena Ferrari, and Andrea Perego. A decentralized security framework for web-based social networks. *International Journal of Information Security and Privacy*, 2(4):22–53, October 2008.

[12] Barbara Carminati, Elena Ferrari, and Andrea Perego. Enforcing access control in web-based social networks. *ACM Transactions on Information and System Security*, 13(1), October 2009.

[13] Dwaine Clarke, Jean-Emile Elien Carl Ellison, Matt Fredette, Alexander Morcos, and Ronald L. Rivest. Certificate chain discovery in SPKI/SDSI. *Journal of Computer Security*, 9(4):285–322, 2001.

[14] David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli. *Role-Based Access Control*. Artech House, 2nd edition, 2007.

[15] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, 4(3):224–274, August 2001.

[16] Philip W. L. Fong, Mohd Anwar, and Zhen Zhao. A privacy preservation model for Facebook-style social network systems. In *Proceedings of the 14th European Symposium on Research In Computer Security (ESORICS'09)*, volume 5789 of *Lecture Notes in Computer Science*, pages 303–320, Saint Malo, France, September. Springer.

[17] Carrie E. Gates. Access control requirements for Web 2.0 security and privacy. In *IEEE Web 2.0 privacy and security workship (W2SP'07)*, Oakland, California, USA, May 2007.

[18] Fausto Giunchiglia, Rui Zhang, and Bruno Crispo. RelBAC: Relation based access control. In *Proceedings of the Fourth International Conference on Semantics, Knowledge and Grid (SKG'08)*, pages 3–11, Beijing, China, December 2008.

[19] Luigi Giuri and Pietro Iglio. Role templates for content-based access control. In *Proceedings of the Second ACM Workshop on Role-Based Access Control (RBAC'97)*, pages 153–159, Fairfax, Virginia, USA, November 1997.

[20] G. Scott Graham and Peter J. Denning. Protection: Principles and practices. In *Proceedings of the 1972 AFIPS Spring Joint Computer Conference*, volume 40, pages 417–429, Alantic City, New Jersey, USA, May 1972.

[21] Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D. Ullman. Protection in operating systems. *Communications of the ACM*, 19(8):461–471, August 1976.

[22] Song hwa Chae and Wonil Kim. Semantic representation of RTBAC: Relationship-based access control model. In *Advances in Web and Network Technologies, and Information Management: Proceedings of APWeb/WAIM 2007 International Workshops: DBMAN 2007, WebETrends 2007, PAIS 2007 and ASWAN 2007*, volume 4537 of *Lecture Notes in Computer Science*, Huang Shan, China, June 2007.

[23] Sebastian Ryszard Kruk, Slawomir Grzonkowski,

Adam Gzella, Tomasz Woroniecki, and Hee-Chul Choi. D-FOAF: Distributed identity management with access rights delegation. In *Proceedings of the First Asian Semantic Web Conference (ASWC'06)*, volume 4185 of *Lecture Notes in Computer Science*, pages 140–154, Beijing, China, September 2006.

[24] Devdatta Kulkarni and Anand Tripathi. Context-aware role-based access control in pervasive computing systems. In *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies (SACMAT'08)*, pages 113–122, Estes Park, CO, USA, June 2008.

[25] Ninghui Li, Benjamin N. Grosof, and Joan Feigenbaum. Delegation logic: A logic-based approach to distributed authorization. *ACM Transactions on Information and System Security*, 6(1):128–171, February 2003.

[26] Ninghui Li, John C. Mitchell, and William H. Winsborough. Design of a role-based trust-management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy (S&P'02)*, pages 114–130, Berkeley, California, USA, May 2002.

[27] Ninghui Li and Mahesh V. Tripunitara. On safety in discretionary access control. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P'05)*, pages 96–109, Oakland, California, USA, May 2005.

[28] Ninghui Li, Mahesh V. Tripunitara, and Ziad Bizri. On mutually exclusive roles and separation-of-duty. *ACM Transactions on Information and System Security*, 10(2), 2007.

[29] Emil Lupu and Morris Sloman. Reconciling role based management and role based access control. In *Proceedings of the Second ACM Workshop on Role-Based Access Control (RBAC'97)*, pages 135–141, Fairfax, Virginia, USA, November 1997.

[30] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 19(2):38–47, February 1996.

[31] Anna Squicciarini, Federica Paci, and Smitha Sundareswaran. PriMa: An effective privacy protection mechanism for social networks. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10)*, pages 320–323, Beijing, China, April 2010.

[32] Anna C. Squicciarini, Mohamed Shehab, and Joshua Wede. Privacy policies for shared content in social network sites. *The VLDB Journal*, 2010. To appear.

[33] Colin Stirling and David Walker. Local model checking in the modal mu-calculus. *Theoretical Computer Science*, 89:161–177, 1991.

[34] Duncan J. Watts. *Small Worlds*. Princeton University Press, 1999.

[35] Stephen Weeks. Understanding trust management systems. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy (S&P'01)*, pages 94–105, Oakland, California, USA, May 2001.