An Access Control Model for Facebook-Style Social Network Systems

Anwar, Mohd; Zhao, Zhen; Fong, Philip W.L.

An Access Control Model for Facebook-Style Social Network Systems

Mohd Anwar
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
manwar@ucalgary.ca

Zhen Zhao
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada
zhao112z@uregina.ca

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

June 23, 2010

Abstract

Recent years have seen unprecedented growth in the popularity of social network systems, with Facebook being an archetypical example. The access control paradigm behind the privacy preservation mechanism of Facebook is distinctly different from such existing access control paradigms as Discretionary Access Control, Role-Based Access Control, Capability Systems, and Trust Management Systems. This work takes a first step in deepening the understanding of this access control paradigm, by proposing an access control model that formalizes and generalizes the access control mechanism of Facebook. The model can be instantiated into a family of Facebook-style social network systems, each with a recognizably different access control mechanism, so that Facebook is but one instantiation of the model. We also demonstrate that the model can be instantiated to express policies that are not currently supported by Facebook, and yet these policies possess rich and natural social significance. Among these policies, we formally identify and characterize a special family of policies known as relational policies, which base their authorization decisions on the dynamic relationship between the resource owner and accessor. We believe the family of relational policies is a unique feature of social network systems. An executable encoding of this model has been developed to support experimentation with various instantiation of our access control model. This work thus delineates the design space of access control mechanisms for Facebook-style social network systems, and lays out a formal framework for policy analysis in these systems.

Keywords Access control, social network system, Facebook, relational policy, formal modeling of protection systems.

1 Introduction

Recent years have seen unprecedented growth in the popularity of ***Social Network Systems (SNSs)***, with stories concerning the privacy and security of such household names as Facebook and MySpace appearing repeatedly in mainstream media. According to boyd and Ellison [17], a "social network site" is characterized by three functions (our paraphrase): (1) these web applications allow users to construct public or semi-public representation of themselves, usually known as user profiles, in a mediated environment; (2) such a site provides formal means for users to articulate their relationships with other users (e.g., friend lists), such that the formal articulation typically reflects existing social connections; (3) users may examine and "traverse" the articulated relationships in order to explore the space of user profiles (i.e., social graph). Identity representation, distributed relationship articulation, and traversal-driven access are thus the defining characteristics of SNSs.

As a user profile contains a constructed representation of the underlying user, the latter must carefully control what contents are visible to whom in her profile in order to preserve privacy. Many existing SNSs offer access control mechanisms that are at best rudimentary, typically permitting coarse-grained, binary visibility control. A pleasant exception is the sophisticated access control mechanism of Facebook. Not only is the Facebook access control mechanism finer grained than many of its competitions, it also offers a wide range of access control abstractions to articulate access control policies, notably abstractions that are based on the topology of the social graph (e.g., the friends-of-friends policy, etc). Unfortunately, this richness comes with a price. By basing access control on the ever-changing topology of the social graph, which is co-constructed by all users of the system, authorization now involves a subtle element of delegation [7, 15] in the midst of discretionary access control [23, 30]. This makes it difficult for users to fully comprehend the privacy consequence of adjusting their privacy settings or befriending other users. A three-pronged research agenda is thus needed to alleviate this problem: (a) understanding the access control paradigm adopted by Facebook, by formally delineating the design space of access control mechanisms induced by this paradigm, (b) articulating the security requirements of SNSs, by formalizing the security properties that should be enforced by systems sharing the same access control paradigm as Facebook, and (c) devising analytical tools to help users assess the privacy consequence of her actions, an endeavor that traditionally belongs to the domain of safety analysis [25, 31, 38], or, more recently, security analysis [29, 30].

This work addresses challenge (a). In particular, this study has two objectives. First, we want to deepen our understanding of the access control paradigm as adopted by Facebook by formally characterizing its distinctiveness. Second, we want to generalize the Facebook access control mechanism, thereby mapping out the design space of access control mechanisms that can potentially be deployed in similar SNSs. To these ends, we have constructed an access control model that captures the access control paradigm of Facebook. The model can be instantiated into a family of Facebook-style SNSs, each with a recognizably different access control mechanism, so that Facebook is but one instantiation of the model. Our contributions are fivefold:

1. Our analysis led us to see the access control mechanism behind Facebook as a form of distributed access control, such that (a) access is mediated by capability-like handles, (b) policies are *intentionally specified* to support delegation, and (c) authorization decision is a function of an abstraction [22] of the global protection state, namely, the social graph.

2. We formalized the above insight into a concrete access control model for delimiting the de-

sign space of access control mechanisms in Facebook-style SNSs. We carefully constrained the information that can be consumed by various elements of the authorization mechanism, so that the only information accessible for the purpose of authorization are local communication history and global acquaintance topology (see Sect. 3). We showed that Facebook is but one instantiation of this model.

3. We demonstrated that the model can be properly instantiated to express a number of access control policies that possess rich and natural social significance: e.g., degree of separation, known quantity, clique, trusted referral, bad company, celebrity, and stranger. The utility of such policies in an information sharing setting is illustrated in a case study. We thus argue that the design space induced by our access control model should be considered in future design of SNSs.

4. We formally characterized a family of SNS access control policies known as *relational policies*, in which authorization decisions are based on how the resource owner and accessor are related in the social graph. We believe this family of policies is a unique feature of Facebook-style SNSs. Our characterization allows one to reason about relational policies in graph-theoretic terms, thereby paving the way to future policy analysis for SNSs.

5. We captured our access control model in a Prolog-based executable encoding. The implementation can be used by researchers to experiment with various instantiations of our model, or can be integrated into an SNS as a policy enforcement engine.

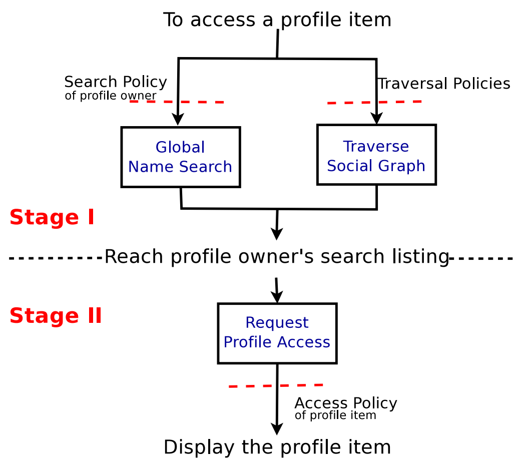
This paper is organized as follows. Sect. 2 provides a high level analysis of the access control mechanism of Facebook, as well as highlights of its distinctiveness and possible generalization. Sect. 3 defines an access control model that captures the above-mentioned distinctiveness and generalization. In Sect. 4, the model is instantiated to mimic the access control mechanism of Facebook, as well as to produce access control policies that are rich in social significance. A formal characterization of relational policies is given in Sect. 5. A case study of modeling an e-learning system as an instantiation of our access control model is provided in Sect. 6. A Prolog-based implementation of our model is described in Sect. 7. Sect. 8 surveys related literature. Sect. 9 suggests future work. Conclusions are given in Sect. 10.

2 Access Control in Facebook and Beyond

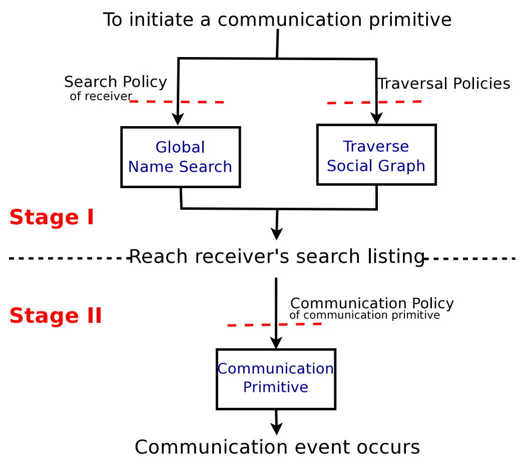
2.1 Access Control in Facebook

We provide here an informal analysis of the Facebook access control mechanism.

Profile and Profile Items. Facebook allows each user to construct a representation of herself in the form of a ***profile***. A profile displays such ***profile items*** as personal information (e.g., favorite books), multimedia contents (e.g., pictures), activity logs (e.g., status), or other user-authored contents (e.g., blog-like postings). Facebook users may grant one another access to the profile items they own.



(a) Accessing profile items.



(b) Initiating communication primitives.

Figure 1: Authorization schemes.

Search Listings and their Reachability. Access to profile items is authorized in two stages (Fig. 1a). In ***Stage I***, the accessing user must ***reach*** the ***search listing*** of the profile owner. Then in ***Stage II***, the accessing user requests access to the profile, and the profile items are selectively displayed. The search listing of a user could be seen as a "capability" [18, 32] of the user in the system, through which access is mediated. There are two means by which a search listing may be reached in Stage I — ***global name search*** and ***social graph traversal***.

Global Name Search. The first means to reach a search listing is to conduct a global name search. A successful search would produce for the accessing user the search listing of the target user. A user may specify a ***search policy*** to allow only a subset of users to be able to reach her search listing through a global name search.

Social Graph Traversal. A second means to reach a search listing is by traversing the ***social graph***. Facebook allows users to articulate their relationships with one another through the construction of ***friend lists***. Every user may list a set of other users as her ***friends***. As friendship is an irreflexive, symmetric binary relation, it induces a simple graph known as the social graph, in which users are nodes and relationships are edges. A user may traverse this graph by examining the friend lists of other users. More specifically, the friend list of a user is essentially the set of search listings of her friends. A user may restrict traversal by specifying a ***traversal policy***, which specifies the set of users who are allowed to examine her friend list after her search listing is reached.

Profile Access. Once the search listing of a profile owner is reached, the accessing user may elect to access the profile, thereby initiating Stage II of authorization. Whether the profile as a whole can be accessed is dictated by another user-specified policy, the details of which we omit[1]. Not every accessing user sees the same profile items when a profile is displayed. The owner may assign

[1]This redundancy is an administrative convenience rather than an essential component of the access control paradigm.

an ***access policy*** to each profile item, dictating who can see that profile item when the profile is accessed. This is the means through which a user may project different representations of herself to different groups of users.

Friendship Articulation and other Communication Primitives. Articulating friendship involves a consent protocol, whereby a user sends a friendship invitation to another user, who may then accept or ignore the invitation. Once a mutual consent is reached, that friendship is recognized by Facebook.

Other than friendship invitation, Facebook also supports other communication primitives, such as messaging, "poking", etc. Common to all these primitives is that the search listing of the receiver must be reached before the communication primitive can be initiated by the sender (Fig. 1b). A user can assign a ***communication policy*** to each communication primitive, specifying the set of users who are allowed to initiate that communication primitive against her once her search listing is reached.

Policies. We have seen in the above discussion that various aspects of user activities are controlled by user-specified policies (e.g., search policy, access policy, etc). This is typical of a discretionary access control system [23, 30], in which a user may grant access privileges to other users. Facebook offers a fixed vocabulary of predefined policies for users to choose from when they are to identify sets of privileged users. As in many capability systems, there is no global name space of users that can be used for the purpose of identifying user sets [32]. Therefore, many of the predefined policies identify user sets indirectly in terms of the topology of the social graph. For example, one may specify that a certain profile item is accessible only by "friends", or that messaging is only available to "friends of friends".

Facebook also defines groups, networks and communities of users so that policies can be formulated in terms of these concepts. We deem user grouping a well-understood concept, and thus focus only on topology-based policies in the sequel.

2.2 Distinctiveness and Generalization

Distinctiveness. Compared with other access control paradigms, the access control paradigm of Facebook is distinctive in at least three ways.

D1 *Capability Mediation.* A common precondition of any access, be it the display of a user profile or the initiation of communication, is the reachability of the search listing of the resource owner (Stage I). This causes user search listings to acquire a role akin to a capability [18, 32]. However, unlike a pure capability system, reachability is necessary but not sufficient for access. Stage-II authorization still consults user-specified policies prior to granting access. Furthermore, Facebook would not be considered by the object capability community to be a pure capability system due to the existence of global name search, a source of ambient authority [32].

D2 *Intensionally Specified Policies.* Due to the lack of a global name space for accessible resources (a common feature in capability systems [32]), privileged users are not specified in policies by names. Instead, they are specified ***intensionally***[2] as the set of users partaking in a certain relationship with the owner of the resource (e.g., friends of friends). Consequently, privileges are not granted to an extensionally specified set of users, as in the case of DAC [23, 30], nor to a centrally administrated set of roles, as in the case of RBAC [37, 21]. Instead, privileges are granted with respect to an intensionally-specified relation. Because the articulation of concrete relationships in that relation are carried out in a distributed and cooperative manner, intensionally specified policies introduce an element of delegation into the access control model of Facebook-style SNSs.

$\mathbf{D3}$ *Abstraction of Communication History into Relationships.* As in many access control systems [39], authorization in Facebook is a function of the history of communication among users (e.g., u invites v to be a friend, v accepts the invitation, and then v is allowed to access resources owned by u). What is special about Facebook is the kind of information that the user-specified policies are allowed to consume. Specifically, the global communication history is abstracted, in the sense of Fong [22], into a social graph that describes the dynamic relationships between users. These relationships become an important basis of authorization decisions.

Perhaps the access control paradigm that is the most comparable to that of Facebook is Trust Management Systems (TMSs) [9, 42]. To fix thoughts, we provide a comparison with the family of TMSs identified by Weeks [42]. We note three points of comparison. First, Weeks' TMSs support the formulation of intentionally specified policies (aka licenses) to avoid the need of centralized identity management. In this respect they share with Facebook a similar style of distributed access control ($\mathbf{D2}$). Second, Facebook is completely mediated, and thus search listing reachability (Stage I) is a precondition of authorization ($\mathbf{D1}$). In contrast, Weeks' TMSs do not control the reachability of principals and their resources. Third, unlike a Weeks' TMSs, Facebook does not base its authorization decision on the exchange of certificates (aka authorizations). Rather, the basis of authorization decision in Facebook is a social graph abstracted from the communication history between users ($\mathbf{D3}$). In our generalization below, this allows us to formulate relational policies that have no analogue in Weeks' TMSs.

Generalization. Facebook embodies the above paradigm of access control ($\mathbf{D1}$–$\mathbf{D3}$) by providing:

$\mathbf{G1}$ a specific protocol for articulating acquaintance, and

$\mathbf{G2}$ a specific vocabulary of access control policies for specifying privileged users.

In the following, we will present a formal model of access control for Facebook-style SNSs, capturing the distinctive paradigm of authorization as identified in $\mathbf{D1}$–$\mathbf{D3}$, while allowing an arbitrary consenting protocol ($\mathbf{G1}$) and policy vocabulary ($\mathbf{G2}$) to be adopted. Therefore, such a model delineates the design space of access control mechanisms embodying such a paradigm.

[2]An extensional definition specifies a concept by enumerating its instances (e.g., $S = \{0, 1, 2\}$). An intensional definition specifies a concept by stating the characteristic property of its instances (e.g., $S = \{x \in \mathbb{N} \mid x < 3\}$).

3 An Access Control Model of Social Network Systems

Notations. We write $\mathbb{N}$ and $\mathbb{B}$ to denote respectively the set of natural numbers and that of boolean values. We identify the two boolean values by 0 and 1. Given a set S, $\mathcal{P}(S)$ is the power set of S, $[S]^k$ is the set of all size-k subsets of S, and, when S is finite and non-empty, $\mathcal{G}(S)$ is the set of all simple graphs with S as the vertex set (i.e., $\mathcal{G}(S) = \{ \langle S, E \rangle \mid E \subseteq [S]^2 \}$). We use the the standard λ-notation for constructing functions [34]: i.e., $(\lambda x \,.\, e)$ is the anonymous function with formal parameter x and body expression e. For example, $(\lambda x \,.\, x^2)$ is a function that returns the square of a given number. We write $S \rightharpoonup T$ for the set of all partial functions with a subset of S as the domain and T as the codomain. Given $f \in S \rightharpoonup T$, $s \in S$, and $t \in T$, we write $f[s \mapsto t]$ to denote the function that is everywhere the same as f except that $f(s) = t$. That is, $f[s \mapsto t] = (\lambda x \,.\, \mathsf{if}\ x = s\ \mathsf{then}\ t\ \mathsf{else}\ f(x))$.

3.1 System

Our model defines a family of Facebook-style SNSs. Every member of the family is a point in the design space of access control mechanisms represented by our model.

3.1.1 Basic Ontology.

A SNS is made up of ***users*** and ***objects*** (aka profile items). Users are members of a finite set Sub. It is assumed that every user owns the same types of objects (e.g., employment information, contact information, etc). Object types are uniquely identified by ***object identifiers***, which are members of a finite set Obj. Consequently, given a user $u \in Sub$ and an object identifier $o \in Obj$, we write $u.o$ to denote the unique type-o object owned by u. When v attempts to access $u.o$, we call v the ***accessor*** and u the ***owner***. Our goal is to model the authorization mechanism by which accessors are granted access to objects. Inspired by Facebook, a SNS consumes two kinds of information in its authorization mechanism — ***communication history*** and ***acquaintance topology***.

3.1.2 Communication History.

Whether one user may access the objects owned by another user depends on their relationship with one another, which in turn is induced by their history of communication. For example, the event of u inviting v to be a friend, and the subsequent event of v accepting the invitation, turn u and v into friends. Such a sequence of events affects if u and v may access the objects of one another. We postulate that a SNS tracks the communication history between every (unordered) pair of users, and bases authorization decisions on this history.

To formalize the above intuition, we postulate that associated with every SNS is a fixed set Σ of ***communication primitives*** (e.g., friendship invitation, acceptance of invitation, etc). A ***communication event*** occurs when one user ***initiates*** a communication primitive and address it to another user.

For the ease of addressing users in the following discussion, we assume, without loss of generality, that the set of users is totally ordered by $\prec$. For each unordered pair of users, $\{u, v\}$, we define an identification function $\iota_{\{u,v\}} : \{u, v\} \to \mathbb{B}$ to be $(\lambda x \,.\, x = \max_{\prec}(u, v))$, where $\max_{\prec}$ returns the greater of its two arguments based on the ordering $\prec$. In other words, the identification function gives a unique Boolean identifier to each of u and v within the unordered pair $\{u, v\}$. The inverse $\iota_{\{u,v\}}^{-1}$ translates Boolean identifiers back to the users they represent. Given a pair of users u and v, a communication event is a member of the set $\mathbb{B} \times \Sigma$, such that the ordered pair (b, a) uniquely identifies the initiator to be $\iota_{\{u,v\}}^{-1}(b)$ and the communication primitive to be a.

Not all communication event sequences are allowed by the SNS. For example, it makes no sense for v to accept a friendship invitation from u when no such invitation has been extended. Built into each SNS is a communication protocol, which constrains the set of event sequences that can be generated at run time. A SNS must ensure that this protocol is honored, and at the same time track communication history for the purpose of authorization. To address both needs, we adopt a minor variant of the security automaton [39] to model the communication protocol between user pairs, as well as to track communication history. We reuse the notational convention in [22]. A ***communication automaton (CA)*** is a quadruple $M = \langle \Sigma, \Gamma, \gamma_0, \delta \rangle$, where Σ is a countable set of communication primitives, Γ is a countable set of ***communication states***, $\gamma_0 \in \Gamma$ is a distinguished ***initial state***, and $\delta : \Gamma \times \mathbb{B} \times \Sigma \rightharpoonup \Gamma$ is a partial ***transition function*** mapping a given current state and a communication event to the next state. Note that, as δ is partial, the next state may not be defined for some argument combinations. In those cases, the automaton gets "stuck", indicating a violation of the communication protocol.

As we shall see in the next section, a SNS tracks, at run time, a mapping $His : [Sub]^2 \to \Gamma$, called the ***global communication state***, which maps each pair of users to their present communication state. The transition function of the communication automaton then dictates the communication events that could occur next between each pair of users. Therefore, the design of a SNS must begin with the specification of a CA.

3.1.3 Acquaintance Topology.

The communication state between a given pair of users is *local* in nature, describing only the communication history between that specific pair of users. Occasionally, an authorization decision may need to consume information that is *global*, involving the communication history of users other than the accessor and owner. Basing authorization decisions on the global communication state (i.e., the mapping His, which records all pair-wise communication states) makes authorization intractable. The global communication state is therefore lifted into an abstract form to facilitate authorization. Specifically, Facebook specifies a symmetric, irreflexive binary relation, ***friendship***, to denote the fact that mutual consent has been reached between two parties in previous communications, to forge an acquaintance relationship with accessibility consequences. Such a binary relation induces a ***social graph***, the global topology of which becomes a second basis for authorization decisions.

Every SNS is equipped with an ***adjacency predicate***, $Adj : \Gamma \to \mathbb{B}$, which translates the communication state between a pair of users into an acquaintance relationship (or the lack thereof). Given an adjacency predicate Adj and the global communication state His, the ***social graph*** is the simple graph formed by the following function:

$$\mathsf{SG}(Adj, His) = \lambda(Adj, His) \,.\, \langle Sub, \{\{u, v\} \in [Sub]^2 \mid Adj(His(\{u, v\}))\}\rangle$$

Intuitively, the vertices of the social graph are the users (Sub), and there is an edge between a pair $\{u, v\}$ of users whenever Adj returns true for the local communication state $His(\{u, v\})$ between

u and v. In the sequel, we will see that the authorization mechanism of a SNS is given no global information other than the social graph, the topology of which can be consulted for authorization decisions.

3.1.4 Policy Predicates.

As mentioned above, a SNS bases its authorization decisions only on two pieces of information: local communication history and global acquaintance topology. We formalize such an information restriction by mandating a specific type signature for the authorization mechanism. Specifically, a ***policy predicate*** is a boolean function with the signature $Sub \times Sub \times \mathcal{G}(Sub) \times \Gamma \to \mathbb{B}$. Given an object owner $u \in Sub$, an object accessor $v \in Sub$, the current social graph $G \in \mathcal{G}(Sub)$, as well as the current communication state $\gamma \in \Gamma$ between the owner and the accessor, a policy predicate returns a boolean value indicating if the access should be granted. Such a predicate has no access to any state information of the SNS other than the arguments, which expose to the authorization process precisely the local communication history and the global acquaintance topology. (See Sect. 4.1 for an example of how local communication history is used in Facebook's authorization mechanism.)

To facilitate presentation, we define policy combinators that allow us to create complex policies from primitive ones. Given policy predicates P_1 and P_2, define $P_1 \vee P_2$ to be the policy predicate $\lambda(u, v, G, \gamma) \,.\, P_1(u, v, G, \gamma) \vee P_2(u, v, G, \gamma)$. The policy predicates $P_1 \wedge P_2$ and $\neg P_1$ can be defined similarly. We also define $\top$ and $\perp$ to be the policy predicates that always return true and false respectively.

3.1.5 User-Specified Policies.

A SNS allows users to specify four types of policies:

1. Every user u may specify a ***search policy*** (i.e., a predicate of the type $Sub \times Sub \times \mathcal{G}(Sub) \times \Gamma \to \mathbb{B}$), which determines if an accessor v is able to produce a search listing of u by performing a global name search of u.

2. Every user u may specify a ***traversal policy***, which determines if an accessor v is able to see the friend list of u once v has reached the search listing of u. If the friend list of u is visible to v, then v will be able to reach the search listings of u's neighbors in the social graph.

3. Every user u may assign a ***communication policy*** for each communication primitive $a \in \Sigma$. Such a policy determines if an accessor v is allowed to initiate communication primitive a with u as the receiver once v has reached u's search listing.

4. Every user u may assign an ***access policy*** to each object identifier $o \in Obj$. This policy specifies if an accessor v may access $u.o$ after reaching u's search listing.

Users may alter the above policies at will. The current settings of these policies thus form part of the run-time state of the SNS.

3.1.6 System.

A Facebook-style SNS, or a ***system*** in short, is an quintuple $N = \langle \mathit{Sub}, \mathit{Obj}, M, \mathit{Adj}, \mathit{PS} \rangle$.

- Sub is a finite set of users.
- Obj is a finite set of object identifiers, so that every object in the system is uniquely identified by an ordered pair in $\mathit{Sub} \times \mathit{Obj}$.
- $M = \langle \Sigma, \Gamma, \gamma_0, \delta \rangle$ is a CA.
- $\mathit{Adj} : \Gamma \rightarrow \mathbb{B}$ is an adjacency predicate.
- $\mathit{PS} = \{\mathit{PS}_r\}_{r \in \mathcal{R}_N}$ is a family of ***policy spaces*** indexed by ***resources*** $r \in \mathcal{R}_N$, such that $\mathcal{R}_N = \{\,\mathsf{search}, \mathsf{traversal}\,\} \cup \Sigma \cup \mathit{Obj}$, and each PS_r is a countable set of policy predicates (i.e., with type signature $\mathit{Sub} \times \mathit{Sub} \times \mathcal{G}(\mathit{Sub}) \times \Gamma \rightarrow \mathbb{B}$). Intuitively, $\mathit{PS}_{\mathsf{search}}$ specifies the set of policy predicates that users may legitimately adopt as their search policies, while $\mathit{PS}_{\mathsf{traversal}}$, PS_a and PS_o specify, respectively, the set of legitimate traversal policies, the set of legitimate communication policies for communication primitive $a \in \Sigma$, and the set of legitimate access policies for object type $o \in \mathit{Obj}$.

Note that users are not free to choose any policy they want. They must select policies built into the system. The design of policy spaces is thus a important component of SNSs.

3.2 System States

3.2.1 State.

Suppose a system $N = \langle \mathit{Sub}, \mathit{Obj}, M, \mathit{Adj}, \mathit{PS} \rangle$ is given such that $M = \langle \Sigma, \Gamma, \gamma_0, \delta \rangle$. Let $\mathcal{R} = \mathcal{R}_N$. A ***state*** of N is a pair $S = \langle \mathit{His}, \mathit{Pol} \rangle$:

- $\mathit{His} : [\mathit{Sub}]^2 \rightarrow \Gamma$ maps each pair of users to their current communication state. Given $\gamma \in \Gamma$, we also define $\mathit{His}_{\langle \gamma \rangle} : [\mathit{Sub}]^2 \cup [\mathit{Sub}]^1 \rightarrow \Gamma$ to be the function $(\lambda \{u, v\} \,.\, \mathsf{if}\ u = v\ \mathsf{then}\ \gamma\ \mathsf{else}\ \mathit{His}(\{u, v\}))$. That is, $\mathit{His}_{\langle \gamma \rangle}$ is the extension of His that maps $\{u, v\}$ to γ whenever $u = v$.
- $\mathit{Pol} : \mathit{Sub} \times \mathcal{R} \rightarrow \bigcup_{r \in \mathcal{R}} \mathit{PS}_r$ is a mapping that records the current policy for every resource of every user. It is required that $\forall u \in \mathit{Sub} \,.\, \forall r \in \mathcal{R} \,.\, \mathit{Pol}(u, r) \in \mathit{PS}_r$.

We model the two stages of authorization as queries against a state. Specifically, these queries model the reachability of search listings and the accessibility of profile items.

3.2.2 Reachability.

Fig. 2 describes the rules for navigating the social graph. Specifically, the sequent "$S \vdash_N v\ \mathsf{finds}\ u$" holds whenever accessor v is permitted to traverse the social graph to reach the search listing of user u. According to Fig. 2, this occurs if one of the following four conditions is satisfied: (a) $v = u$ (F-SLF); (b) v is adjacent to u in the social graph (F-FRD); (c) v may recursively reach a neighbor u' of u, and the traversal policy of u' allows v to access the friend list of u' (F-TRV); (d)

$$S \vdash_N u \ \mathsf{finds}\ u \tag{F-SLF}$$

$$\frac{N = \langle \_, \_, \_, \mathit{Adj}, \_ \rangle \qquad G = \mathsf{SG}(\mathit{Adj}, \mathit{His}) \qquad \{u, v\} \in E(G)}{\langle \mathit{His}, \mathit{Pol} \rangle \vdash_N v \ \mathsf{finds}\ u} \tag{F-FRD}$$

$$\frac{\begin{array}{c}\langle \mathit{His}, \mathit{Pol} \rangle \vdash_N v \ \mathsf{finds}\ u' \\ N = \langle \_, \_, M, \mathit{Adj}, \_ \rangle \qquad M = \langle \_, \_, \gamma_0, \_ \rangle \qquad \gamma = \mathit{His}_{\langle \gamma_0 \rangle}(\{u', v\}) \\ G = \mathsf{SG}(\mathit{Adj}, \mathit{His}) \qquad \{u, u'\} \in E(G) \qquad \mathit{Pol}(u', \mathsf{traversal})(u', v, G, \gamma)\end{array}}{\langle \mathit{His}, \mathit{Pol} \rangle \vdash_N v \ \mathsf{finds}\ u} \tag{F-TRV}$$

$$\frac{\begin{array}{c}N = \langle \_, \_, M, \mathit{Adj}, \_ \rangle \qquad M = \langle \_, \_, \gamma_0, \_ \rangle \qquad \gamma = \mathit{His}_{\langle \gamma_0 \rangle}(\{u, v\}) \\ G = \mathsf{SG}(\mathit{Adj}, \mathit{His}) \qquad \mathit{Pol}(u, \mathsf{search})(u, v, G, \gamma)\end{array}}{\langle \mathit{His}, \mathit{Pol} \rangle \vdash_N v \ \mathsf{finds}\ u} \tag{F-SCH}$$

Figure 2: Definition of the reachability sequent $S \vdash_N v$ finds u.

$$\frac{\begin{array}{c}\langle \mathit{His}, \mathit{Pol} \rangle \vdash_N v \ \mathsf{finds}\ u \\ N = \langle \_, \_, M, \mathit{Adj}, \_ \rangle \qquad M = \langle \_, \_, \gamma_0, \_ \rangle \qquad \gamma = \mathit{His}_{\langle \gamma_0 \rangle}(\{u, v\}) \\ G = \mathsf{SG}(\mathit{Adj}, \mathit{His}) \qquad \mathit{Pol}(u, o)(u, v, G, \gamma)\end{array}}{\langle \mathit{His}, \mathit{Pol} \rangle \vdash_N v \ \mathsf{reads}\ u.o} \tag{R-ACC}$$

Figure 3: Definition of the accessibility sequent $S \vdash_N v$ reads $u.o$.

the search policy of u permits v to reach her through global name search (F-SCH). As we shall see, reachability is a necessary condition for access (i.e., Stage-I authorization). Properly controlling the reachability of ones search listing is an important component of protection.

3.2.3 Accessibility.

Fig. 3 specifies the rules for object access. Specifically, the sequent "$S \vdash_N v$ reads $u.o$" holds whenever accessor v is permitted to access object o of owner u. According to Fig. 3, access is permitted if v can reach the search listing of u, and the access policy of u allows access (R-ACC).

3.3 State Transition

The state of a system is changed by a set of transition rules. To allow us to refer to these transitions, we define a set $\mathcal{T}_N$ of transition identifiers, the syntax of which is given in Fig. 4. The convention is that the first argument of a constructor is always the initiator of the transition. We write $\mathit{initiator}(t)$ for the initiator of transition identifier t.

Fig. 5 defines the state transition relation, $S \xrightarrow{t}_N S'$, which specifies when a transition identified by t may occur from state S to state S'. Rule T-HIS specifies the effect of communication events. It ensures that accessor v may communicate with user u only when (a) v reaches u, (b) the communication event honors the communication protocol of the system, and (c) the specific communication primitive initiated by v is permitted by the communication policy of u. If all three preconditions are satisfied, then the communication state of the two users will change according

$$\begin{array}{lcll} \mathcal{T}_N \ni t & ::= & \mathsf{com}(v,u,a) & \text{for } u,v \in \mathit{Sub},\, a \in \Sigma \\ & | & \mathsf{pol}(u,r,P) & \text{for } u \in \mathit{Sub},\, r \in \mathcal{R}_N,\, P \in \mathit{PS}_r \end{array}$$

Figure 4: Definition of the set $\mathcal{T}_N$ of transition identifiers for a system $N = \langle \mathit{Sub}, \mathit{Obj}, M, \mathit{Adj}, \mathit{PS} \rangle$, where $M = \langle \Sigma, \Gamma, \gamma_0, \delta \rangle$.

$$\frac{\begin{array}{c} u \neq v \qquad \langle \mathit{His}, \mathit{Pol} \rangle \vdash_N v \text{ finds } u \\ N = \langle \_, \_, M, \mathit{Adj}, \_ \rangle \qquad M = \langle \_, \_, \_, \delta \rangle \qquad G = \mathsf{SG}(\mathit{Adj}, \mathit{His}) \\ \gamma = \mathit{His}(\{u,v\}) \qquad b = \iota_{\{u,v\}}(v) \qquad \gamma' = \delta(\gamma, b, a) \\ \mathit{Pol}(u,a)(u,v,G,\gamma) \qquad \mathit{His}' = \mathit{His}[\{u,v\} \mapsto \gamma'] \end{array}}{\langle \mathit{His}, \mathit{Pol} \rangle \xrightarrow{\mathsf{com}(v,u,a)}_N \langle \mathit{His}', \mathit{Pol} \rangle} \quad \text{(T-COM)}$$

$$\frac{N = \langle \_, \_, \_, \_, \{\mathit{PS}_r\}_{r \in \mathcal{R}_N} \rangle \qquad P \in \mathit{PS}_r \qquad \mathit{Pol}' = \mathit{Pol}[(u,r) \mapsto P]}{\langle \mathit{His}, \mathit{Pol} \rangle \xrightarrow{\mathsf{pol}(u,r,P)}_N \langle \mathit{His}, \mathit{Pol}' \rangle} \quad \text{(T-POL)}$$

Figure 5: Definition of the state transition relation $S \xrightarrow{t}_N S'$.

to the communication protocol of the system. Rule (T-POL) specifies change of policies. The rule ensures that the policy predicate selected by the initiating user for a given resource belongs to the corresponding policy space of that resource. We write $S \xrightarrow{w}_N S'$ for $w \in (\mathcal{T}_N)^*$ whenever S can transition to S' through the sequence of transitions identified by w.

4 Sample Instantiations

We illustrate the utility of our model by considering concrete instantiations.

4.1 Facebook as an Instantiation

We begin with an instantiation of the model to *mimic* the access control mechanism of Facebook. We explicitly eschew claiming that the instantiation accurately mirrors the access control mechanism of Facebook. Aiming for accuracy is inevitably futile because the Facebook technology is a moving target. Instead, our goal is to verify that our model captures the essential features of Facebook's access control mechanism, although it does not necessarily mirrors every details of that mechanism.

Consider the SNS $\mathcal{FB}_{lite} = \langle \mathit{Sub}, \mathit{Obj}, M, \mathit{Adj}, \mathit{PS} \rangle$ defined as follows. Sub is the set of all user identifiers. Obj is the set of the profile item names, say, $\{$ Basic-Information, Contact-Information, Personal-Information, Status-Updates, Wall-Posts, Education-Info, Work-Info $\}$.

The communication automaton $M = \langle \Sigma, \Gamma, \gamma_0, \delta \rangle$ is defined such that $\Sigma = \{\mathsf{invite}, \mathsf{accept}, \mathsf{ignore}, \mathsf{remove}\}$, $\Gamma = \{\mathsf{stranger}, \mathsf{invited\text{-}1}, \mathsf{invited\text{-}0}, \mathsf{friend}\}$, $\gamma_0 = \mathsf{stranger}$, and δ is defined as in Fig. 6.

The adjacency predicate Adj is $(\lambda\gamma \,.\, \gamma = \mathsf{friend})$.

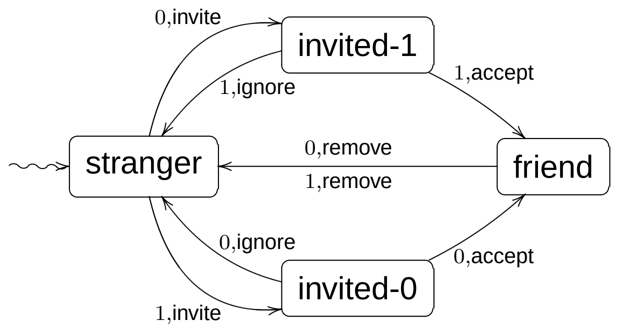


Figure 6: Transition diagram for the communication automaton of $\mathcal{FB}_{lite}$.

Policy	**Semantics**
no-one	$\bot$
only-me	$\lambda(u, v, G, \gamma) \,.\, u = v$
only-friends	only-me $\vee$ $(\lambda(u, v, G, \gamma) \,.\, \{u, v\} \in E(G))$
friends-of-friends	only-friends $\vee$ $(\lambda(u, v, G, \gamma) \,.\, (\exists v' \in Sub \,.\, \{u, v'\} \in E(G) \wedge \{v', v\} \in E(G)))$
everyone	$\top$

Figure 7: A list of Facebook-inspired policy predicates.

The traversal policy space is $PS_{\mathsf{traversal}} = \{\mathsf{no\text{-}one}, \mathsf{only\text{-}me}, \mathsf{only\text{-}friends}, \mathsf{friends\text{-}of\text{-}friends}, \mathsf{everyone}\}$, where the policy predicates are defined in Fig. 7.

The search policy space PS_{search} could have been defined in the same way as $PS_{\mathsf{traversal}}$ had it not been the following complication. Once v extends a friendship invitation to u, the search listing of v will become accessible from u. Rather than introducing additional complexities into the model, we tailor the search policy of u to allow this behavior. To this end, the following policy predicate is introduced:

$$\mathsf{owner\text{-}invited} = (\lambda(u, v, G, \gamma) \,.\, (u \prec v \wedge \gamma = \mathsf{invited\text{-}1}) \vee (v \prec u \wedge \gamma = \mathsf{invited\text{-}0}))$$

This predicate returns true iff u has extended a friendship invitation to v. Then PS_{search} is defined as $\{P \vee \mathsf{owner\text{-}invited} \mid P \in PS_{\mathsf{traversal}}\}$. As a result, initiating a friendship invitation will cause the search listing of the initiator to become accessible to the invited party. This illustrates how local communication history can be used in authorization.

For a typical $o \in Obj$, the access policy space PS_o can be defined to be the same as $PS_{\mathsf{traversal}}$. The only exception is that, once u sends a friendship invitation to v, some distinguished objects of u, say Basic-Information, would become accessible to v. We therefore set $PS_{\mathsf{Basic\text{-}Information}} = PS_{\mathsf{search}}$.

The communication policy space is defined as follows:

$$PS_a = \begin{cases} \{\,\mathsf{no\text{-}one}, \mathsf{friends\text{-}of\text{-}friends}, \mathsf{everyone}\,\} & \text{if } a = \mathsf{invite} \\ \{\,\mathsf{everyone}\,\} & \text{otherwise} \end{cases}$$

First, note that the communication automaton M already specifies in what communication state is a given communication primitive applicable. There is no need for tailoring policies for enforcing applicability constraints. That is why $PS_a = \{\textsf{everyone}\}$ for most a. Secondly, a user may not always want to allow friendship invitations from strangers. $PS_{\textsf{invite}}$ is therefore set to $\{\textsf{no-one}, \textsf{friends-of-friends}, \textsf{everyone}\}$.

Limitations We are fully aware that $\mathcal{FB}_{lite}$ does not capture all aspects of the access control mechanism of Facebook. Some missing features include:

- Groups, networks, communities, alternative friend lists and blocking are not modeled.
- Obj does not cover the full list of profile item types in Facebook.
- The communication protocol does not support poking, messaging, and other minor user interactions.
- The communication automaton is intentionally simplified to omit minor transitions, including, for example, friendship establishment due to mutual invitation.
- There is a "closed-world assumption" in our model, in the sense that all interactions of the model with the outside world is ignored. Facebook, however, interacts with the outside world in subtle ways: e.g., the profile could be made searchable by web search engine.

Nevertheless $\mathcal{FB}_{lite}$ illustrates how the model can be instantiated. Reasonable efforts will allow one to capture more aspects of Facebook in this model. For example, a group or a network could be modeled as a virtual user. Group membership could then be modeled as friendship between a group member and the virtual user that represents the group. A policy similar to friends-of-friends will allow fellow group members to access objects owned by one another.

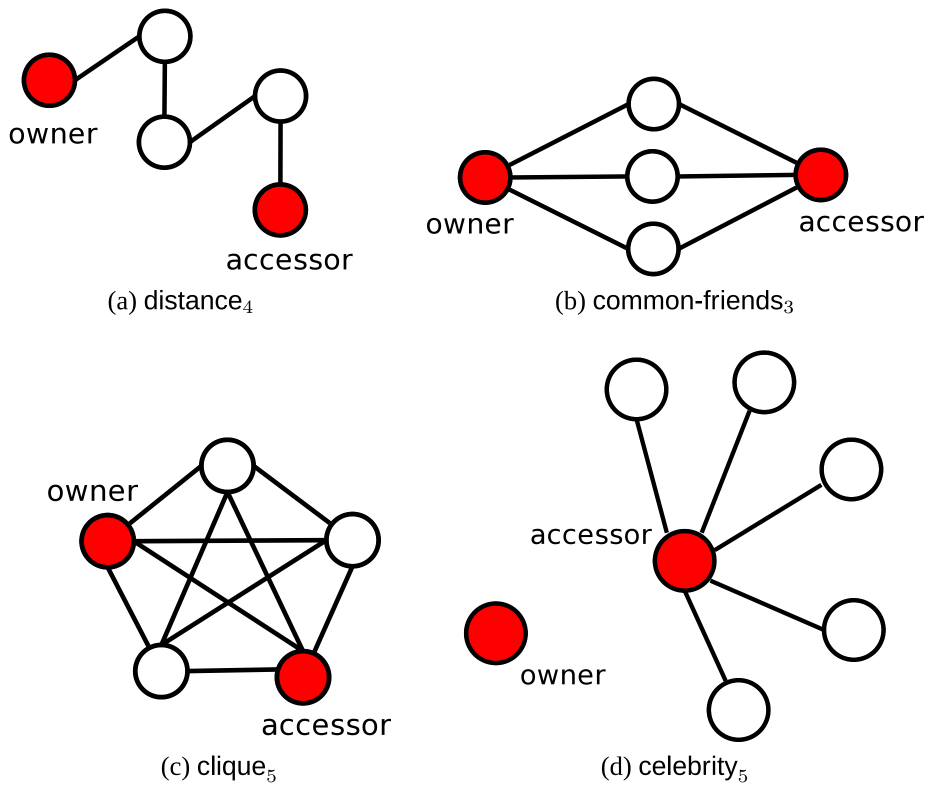
4.2 A Parade of Policy Predicates

This section explores policy predicates other than those already offered by Facebook. The goal is to illustrate the possibilities supported by the proposed model. It is assumed that adjacency in the social graph is induced by some from of social acquaintance (e.g., friendship), which in turn is formed by a mutual consent protocol (e.g., friendship invitation and acceptance). To help fix thoughts, assume that the following policies are used as access policies (although they are equally applicable as communication, traversal, and search policies):

Degree of Separation. For $k \geq 1$, let policy $\textsf{distance}_k$ to be the following predicate:

$$\lambda(u, v, G, \gamma) \;.\; d_G(u, v) \leq k$$

where $d_G(u, v)$ denotes the distance between vertices u and v in graph G. See Fig. 8a for a scenario in which such a policy is satisfied. This policy allows user v to access an object of user u when the distance between u and v in the social graph G is no more than k. This is a straightforward generalization of Facebook's friends-of-friends to an arbitrary degree of separation. Objects are granted not only to friends, but also to individuals within a "social circle" of radius k. Here, the distance between two nodes in the social graph is considered a quantitative measure of the degree of acquaintance.



(a) distance$_4$ (b) common-friends$_3$ (c) clique$_5$ (d) celebrity$_5$

Figure 8: Scenarios satisfying sample policies.

Known Quantity. For $k \geq 1$, let policy $\textsf{common-friends}_k$ be the following predicate:

$$\textsf{only-friends} \vee (\lambda(u, v, G, \gamma) \,.\, |N_G(u) \cap N_G(v)| \geq k)$$

where $N_G(u)$ is the ***neighborhood*** of u in graph G, which is defined to be the vertex set $\{v \in V(G) \mid \{u, v\} \in E(G)\}$. See Fig. 8b for a scenario satisfying this policy. Intuitively, the policy permits access between a pair of distinct users when they share at least k common friends. This is another generalization of Facebook's friends-of-friends to an arbitrary number of intermediaries. Access is granted when an enough number of friends know the person. That is, the person is a "known quantity" among friends. Here, the number of common friends becomes a fine-grained quantitative measure of the degree of acquaintance for friends of friends. Note that $\textsf{common-friends}_1 = \textsf{distance}_2$.

Clique. For $k \geq 2$, define policy $\textsf{clique}_k$ as follows:

$$\textsf{only-me} \vee (\lambda(u, v, G, \gamma) \,.\, (\exists G' \,.\, G' \subseteq G \wedge G' \cong K_k \wedge \{u, v\} \subseteq V(G')))$$

where $G_1 \subseteq G_2$ iff graph G_1 is a subgraph of graph G_2, $G_1 \cong G_2$ iff graph G_1 is isomorphic to graph G_2, and K_k is the complete graph of order k. In short, access is granted when u and v belong to a k-clique (Fig. 8c). The intuition is that if two individuals are both part of a tightly-knit group, in which everyone knows everyone else, then the two must know each other very well, and thus access can be safely granted. Here, the size of the largest clique to which two friends belong is used as a fine-grained quantitative measure of the degree of acquaintance of friends. Note that $\textsf{clique}_2 = \textsf{distance}_1$.

Trusted Referral. Given $k \geq 1$ and $U \subseteq Sub$, let policy $\textsf{common-friends}_{k,U}$ be the following predicate:

$$\textsf{only-friends} \vee (\lambda(u, v, G, \gamma) \,.\, |N_G(u) \cap N_G(v) \cap U| \geq k)$$

The policy grants access whenever v is a mutual friend of at least k users belonging to a specific user set U. Essentially, friends in U are considered more trusted than others in mediating access. Acquaintance with them becomes a license to access. Note that $\textsf{common-friends}_{k,Sub} = \textsf{common-friends}_k$.

Bad Company. Given $k \geq 0$ and $U \subseteq Sub$, define policy $\textsf{bad-company}_{k,U}$ as follows:

$$\lambda(u, v, G, \gamma) \,.\, (|N_G(v) \cap U| \leq k)$$

This policy denies access by the demerits of association with users in U. The policy allows access when the accessor v is a friend of no more than k members of a "black-listed" user set U. Intuitively, the members of U are considered "bad influence", and association with them reduce the trust that u may have on v. In particular, when $k = 0$, u denies v access whenever the latter is associated with any of the users in U.

Celebrity. Given $k \geq 1$, let policy $\textsf{celebrity}_k$ be defined as:

$$\lambda(u, v, G, \gamma) \,.\, (\deg_G(v) \geq k)$$

The policy allows or denies access based on the degree $\deg_G(v)$ of the accessor node v, where $\deg_G(v) = |N_G(v)|$, i.e., the number of edges the accessor node is incident to. Such a policy grants access only to users who are "popular". See Fig. 8d for a scenario in which such a policy is satisfied.

Stranger. Consider $\neg\textsf{distance}_k$, the negation of $\textsf{distance}_k$. Such a policy allows access when the distance between two parties is more than k. The intention is to offer access to objects reserved for "strangers".

5 A Theory of Relational Policies

Access control systems found in SNSs in general, and Facebook-style SNSs in particular, support a unique style of access control policies. These policies authorize by examining the dynamic relationship between the owner and the accessor. We call such policies ***relational policies***. Relational policies are the natural consequence of the design features **D2** (intensionally specified policies) and **D3** (abstraction of communication history into relationships) of Facebook-style SNS. The goal of this section is to offer a formal characterization of this interesting policy class in the context of Facebook-style SNSs, thereby offering a deeper understanding of this intuitive notion of relational policies.

5.1 Topology-Based Policies

There is a fundamental difference between a policy such as $\mathsf{only\text{-}friends} \vee \mathsf{owner\text{-}invited}$ from the search policy space PS_{search} of $\mathcal{FB}_{lite}$ (Sect. 4.1), and a policy such as $\mathsf{common\text{-}friends}_k$ (Sect. 4.2). In the former case, the authorization decision is based partly on the *local communication history* between the accessor and the owner (i.e., the disjunct owner-invited), whereas, in the latter case, the topology of the social graph is the sole basis of the authorization decision. A careful reader will also notice a subtle difference between $\mathsf{common\text{-}friends}_k$ and $\mathsf{common\text{-}friends}_{k,U}$: $\mathsf{common\text{-}friends}_k$ consumes only the topological information of the current social graph, while $\mathsf{common\text{-}friends}_{k,U}$ depends on the *identity* of the users in U. A policy predicate that makes authorization decisions by consulting only topological information of the social graph, and eschews dependence on history or identity information, is of particular interest. Such policies highlights one of the uniqueness es of the access control paradigm behind SNSs: authorization decisions depend solely on the current topology of the social graph, but not on who the accessor is (e.g., RBAC), or what the accessor has done in the past (i.e., History-Based Access Control). This section offers a characterization of these policies.

Definition 1. *A* ***birooted graph*** $G_{(u,v)}$ *is a triple* $\langle G, u, v\rangle$ *such that* G *is a simple graph and* $u, v \in V(G)$*. The* ***roots*** u *and* v *need not be distinct. We write* $\mathcal{B}(S) = \{G_{(u,v)} \mid G \in \mathcal{G}(S)\}$ *to denote the set of all birooted graphs with vertices coming from a set* S*.*

$G'_{(u,v)}$ *is a* ***(birooted) subgraph*** *of* $G_{(u,v)}$*, written as* $G'_{(u,v)} \subseteq G_{(u,v)}$*, iff* G' *is a subgraph of* G*. (Note the matching roots.) We also write* $G'_{(u,v)} \subset G_{(u,v)}$*, iff* G' *is a proper subgraph of* G*. We say that* $G_{(u,v)}$ *and* $G'_{(u',v')}$ *are* ***isomorphic*** *iff there exists a graph isomorphism* $\phi : V(G) \to V(G')$ *between* G *and* G'*, such that* $\phi(u) = u'$ *and* $\phi(v) = v'$*. In this case we write* $G_{(u,v)} \cong_\phi G'_{(u',v')}$ *or simply* $G_{(u,v)} \cong G'_{(u',v')}$*. We also write* $G'_{(u',v')} \lesssim G_{(u,v)}$*, and say that* $G_{(u,v)}$ ***contains*** $G'_{(u',v')}$ *whenever there is a birooted graph* $G''_{(u,v)}$ *such that* $G''_{(u,v)} \subseteq G_{(u,v)}$ *and* $G'_{(u',v')} \cong G''_{(u,v)}$*.*

Other than the above cases, when we apply graph-theoretic languages to birooted graphs, we are essentially referring to the underlying simple graphs.

Definition 2. *A policy predicate* P *is* ***topology-based*** *iff, for every users* u*,* v*,* u' *and* v'*, every social graphs* G *and* G'*, and every communication states* γ *and* γ'*, we have:*

$$G_{(u,v)} \cong G'_{(u',v')} \ \Rightarrow\ P(u, v, G, \gamma) = P(u', v', G', \gamma')$$

For a topology-based policy, neither the local communication history between the owner and the accessor nor their identities are taken into consideration in authorization. In short, a topology-based policy is a birooted version of a graph property [19]. Consequently, $\mathsf{only\text{-}friends} \vee \mathsf{owner\text{-}invited}$ is not topology-based, because its value depends on the local communication history between the owner and the accessor. Similarly, neither $\mathsf{common\text{-}friends}_{k,U}$ nor $\mathsf{bad\text{-}company}_{k,U}$ is topology-based, because their values depend on the identities of the users in U. The first column of the table in Figure 9 summarizes this classification.

The class of topology-based policies is closed under boolean conjunction, disjunction and negation. That is, given topology-based policies P_1 and P_2, the following are also topology-based: $P_1 \wedge P_2$, $P_1 \vee P_2$, and $\neg P_1$. Note also that both $\top$ and $\bot$ are trivially topology-based.

	topology-based	local	monotonic	anti-monotonic
$\top$	yes	yes	yes	yes
$\bot$	yes	yes	yes	yes
$\mathsf{distance}_k$	yes	yes	yes	no
$\mathsf{common\text{-}friends}_k$	yes	yes	yes	no
clique_k	yes	yes	yes	no
$\mathsf{only\text{-}friends} \lor \mathsf{owner\text{-}invited}$	no	yes	yes	no
$\mathsf{common\text{-}friends}_{k,U}$	no	yes	yes	no
$\mathsf{bad\text{-}company}_{k,U}$	no	no	no	yes
$\mathsf{celebrity}_k$	yes	no	yes	no
$\mathsf{celebrity}_k \land \mathsf{distance}_l$	yes	yes	yes	no
$\neg\mathsf{distance}_k$	yes	yes	no	yes

Figure 9: Classification of sample policy predicates.

5.2 Local Policies and Relational Policies

The definition of topology-based policies rules out reliance on history and identity information in authorization. The definition, however, does not precisely pin point the particular kind of topological information that relational policies should rely on. Specifically, we want to express the intuitive idea that a relational policy describes a particular way in which the owner and the accessor are "related." The following definition is designed to express exactly this notion.

Definition 3. *A policy predicate P is said to be* ***local*** *iff, for every $u, v \in Sub$, $G \in \mathcal{G}(Sub)$, $e \in [Sub]^2$, and $\gamma \in \Gamma$, we have*

$$P(u, v, G + e, \gamma) \neq P(u, v, G, \gamma) \;\Rightarrow\; u,\, v \text{ and } e \text{ belong to the same component in } G + e$$

where $G+e$ denotes the graph obtained by adding an extra edge e into graph G. A policy predicate that is not local is said to be ***global****.*

In short, the removal of an edge can alter the authorization decision of a local policy only if that edge belongs to the common component[3] of the owner and accessor. Local-ness captures the intuition that certain policies specify properties of the extended neighborhood shared by the owner and the accessor, and that a necessary condition for access is that the owner and the accessor are "related" in some manner. Local policies are also of interest to us because of tractability considerations. Evaluation of a global policy predicate could potentially involve examining the entire social graph. Evaluation of a local policy predicate involves only scanning the immediate neighborhood of either the owner or the accessor.

The second column of Figure 9 gives examples of local and global policies. Almost all the policies we have considered are local, with $\mathsf{bad\text{-}company}_{k,U}$ and $\mathsf{celebrity}_k$ as the only exceptions.

[3]A component of a simple graph is a maximal connected subgraph.

Their evaluation depends only on the neighborhood of the accessor, but not whether the owner and the accessors are related in some specific way. In other words, they capture a "property" of the accessor rather than a "relationship" between the accessor and the owner. Note that, in some cases, the conjunction of a global policy with a local policy results in a local one, as is in the case of $\mathsf{distance}_k \wedge \mathsf{celebrity}_l$.

Local-ness is preserved by $\wedge$, $\vee$ and $\neg$. The constant policies $\top$ and $\bot$ are both trivially local. The definition of local-ness is orthogonal to that of topology-based policies, as is already hinted by the examples in Figure 9.

We now have all the devices we need for formalizing relational policies.

Definition 4. *A policy predicate is **relational** iff it is both topology based and local.*

This definition formalizes the intuition that relationality is independent of identity and access history, but instead is a reflection of a specific kind of relationship between the owner and the accessor. Again, relationality is preserved by the three boolean connectives, and the constant policies are trivially relational.

5.3 Monotonicity and Anti-monotonicity

The rest of this section provides an alternative characterization for certain relational policies. This characterization offers insight to the nature of relationality, and provides a means for us to analyze relational policies. To this end, we examine two additional ways by which policy predicates can be categorized.

Definition 5. *A policy predicate P is **monotonic** iff*

$$P(u, v, G, \gamma) \;\Rightarrow\; P(u, v, G + e, \gamma)$$

*for every $u, v \in Sub$, $G \in \mathcal{G}(Sub)$, $e \in [Sub]^2$, and $\gamma \in \Gamma$. Conversely, a policy predicate P is **anti-monotonic** iff*

$$P(u, v, G + e, \gamma) \;\Rightarrow\; P(u, v, G, \gamma)$$

for every $u, v \in Sub$, $G \in \mathcal{G}(Sub)$, $e \in [Sub]^2$, and $\gamma \in \Gamma$.

Under a monotonic policy, adding edges into the social graph never disables access, and removing edges never enables access. Monotonic policies are therefore used for reserving access to "closely-related" users. Under an anti-monotonic policy, access becomes more difficult as the social graph becomes denser. Anti-monotonic policies are therefore used usually for preserving privacy: disclosure of information only to those who do not know you well.

Sometimes privacy is acquired beyond exclusive intimacy. Rubin observed that something is easier to disclose to strangers than to close friends and family and termed it as "passing stranger effect" [36]. For the same reason, a teenager would resist to constant monitoring by her parents even when she trusts and discloses to her parents the most. Therefore, to achieve privacy, self-disclosure has to be targeted differently for someone closely related (more connected) vs. someone at distant (less connected). While monotonic policies grant access to someone more connected, anti-monotonic policies grant access to someone at distant.

Monotonic policies are also interesting because its evaluation involves ascertaining the presence of certain social graph edges rather than their absence. This sole dependence on positive information makes it possible to check monotonic policies in a distributed setting, where complete knowledge of the social graph may not be possible (see [14] for a distributed implementation of SNS).

Figure 9 provides examples of monotonic and anti-monotonic policies. Almost all the policies we have considered are monotonic, with the exceptions of $\textsf{bad-company}_{k,U}$ and $\neg\textsf{distance}_k$. These two policies are anti-monotonic: removing edges from the social graph could lead to the enabling of access.

Both monotonicity and anti-monotonicity are preserved by the positive policy combinators $\wedge$ and $\vee$. As expected, $\neg P$ is anti-monotonic if P is monotonic, and vice versa. The constant predicates $\top$ and $\bot$ are the only policies that are both monotonic and anti-monotonic. Lastly, note that the definition of monotonicity and anti-monotonicity is orthogonal to that of topology-based or local policies, as is already hinted by the examples in Figure 9.

5.4 The Characterization Theorem

To deepen our understanding of relational policies, we provide here an alternative characterization of these policies.

Definition 6. *The policy predicate positively induced by a set $\mathcal{B} \subseteq \mathcal{B}(Sub)$ is the following:*

$$P^+_{\mathcal{B}} = \lambda(u, v, G, \gamma) \,.\, (\exists\, G'_{(u',v')} \in \mathcal{B} \,.\, G'_{(u',v')} \lesssim G_{(u,v)})$$

Dually, the following is the policy predicate negatively induced by $\mathcal{B}$:

$$P^-_{\mathcal{B}} = \lambda(u, v, G, \gamma) \,.\, (\forall\, G'_{(u',v')} \in \mathcal{B} \,.\, G'_{(u',v')} \not\lesssim G_{(u,v)})$$

Proposition 7. $P^-_{\mathcal{B}} = \neg P^+_{\mathcal{B}}$.

The validity of the proposition above follows directly from the definitions of $P^-_{\mathcal{B}}$ and $P^+_{\mathcal{B}}$.

Example 8. *The following are examples of positively and negatively induced policy predicates. (We assume that there is a total ordering of the vertices in the vertex set Sub: u_0, u_1, u_2, $\ldots$)*

- $\top = P^-_{\emptyset} = P^+_{\mathcal{B}(Sub)} = P^+_{\mathcal{B}^\star}$*, where* $\mathcal{B}^\star = \{G^0{}_{(u_0,u_0)}, G^0{}_{(u_0,u_1)}\}$*, and* $G^0 \in \mathcal{G}(Sub)$ *is an empty graph (i.e.,* G^0 *has no edge).*
- $\bot = P^+_{\emptyset} = P^-_{\mathcal{B}(Sub)} = P^-_{\mathcal{B}^\star}$*, where* $\mathcal{B}^\star = \{G^0{}_{(u_0,u_0)}, G^0{}_{(u_0,u_1)}\}$*, and* G^0 *is defined as above.*
- $\textsf{distance}_k = P^+_{\mathcal{B}}$*, where* $\mathcal{B} = \{P^0{}_{(u_0,u_0)}, P^1{}_{(u_0,u_1)}, \ldots, P^k{}_{(u_0,u_k)}\}$*, and the edges of* $P^i \in \mathcal{G}(Sub)$ *form a length-*i *path* $u_0u_1 \ldots u_i$*.*
- $\textsf{clique}_k = P^+_{\mathcal{B}}$*, where* $\mathcal{B} = \{G^0{}_{(u_0,u_0)}, C^k{}_{(u_0,u_1)}\}$*,* G^0 *is defined as above, and the edges of* $C^k \in \mathcal{G}(Sub)$ *form a* k*-clique on vertices* u_0*,* u_1*, …,* u_{k-1}*.*

The following is a characterization for monotonic relational policies[4].

[4]In the statement of Theorem 9, by saying that an edge is connected to a vertex we mean that there is a path between an end of the edge and the vertex.

Theorem 9. *A policy predicate P is both topology based and monotonic iff it is positively induced by a set $\mathcal{B}$ of birooted graphs. Moreover, P is local iff $\mathcal{B}$ can be constructed in such a way that every edge of every birooted graph in $\mathcal{B}$ is connected to the two roots.*

Proof. We begin with the first statement of the theorem. The if direction ($\Leftarrow$) is immediate. We consider the only-if direction ($\Rightarrow$). Let $\prec$ be any lexicographical ordering of $\mathcal{B}(Sub)$. Construct $\mathcal{B} = \mathcal{B}' \cap \mathcal{B}''$, where

$$\mathcal{B}' = \{\, G_{(u,v)} \in \mathcal{B}(Sub) \mid P(u,v,G,\gamma_0) \wedge \\ \neg\, \exists\, G'_{(u,v)} \in \mathcal{B}(Sub)\,.\, G'_{(u,v)} \subset G_{(u,v)} \wedge P(u',v',G',\gamma_0)\,\}$$

and

$$\mathcal{B}'' = \{\, G_{(u,v)} \in \mathcal{B}(Sub) \mid P(u,v,G,\gamma_0) \wedge \\ \neg\, \exists\, G'_{(u',v')} \in \mathcal{B}(Sub)\,.\, G'_{(u',v')} \prec G_{(u,v)} \wedge G'_{(u',v')} \cong G_{(u,v)} \wedge P(u',v',G',\gamma_0)\,\}$$

Intuitively, $\mathcal{B}'$ contains those birooted graphs $G_{(u,v)}$ satisfying P at γ_0, such that one cannot find a proper subgraph of $G_{(u,v)}$ that also satisfies P at γ_0. Similarly, $\mathcal{B}''$ contains those birooted graphs $G_{(u,v)}$ satisfying P at γ_0, such that $G_{(u,v)}$ is lexicographically the smallest birooted graph among the isomorphic images of $G_{(u,v)}$ to satisfy P at γ_0. $\mathcal{B}$ contains graphs that are minimal in both senses. We want to show that $P = P^+_{\mathcal{B}}$.

Suppose $P(u,v,G,\gamma)$. Then, $P(u,v,G,\gamma_0)$ because P is topology based. Consider the edge-minimal subgraph G' of G such that $P(u,v,G',\gamma_0)$ holds (edge-minimal in the sense that either G' contains no edge, or the removal of any edges from G' renders P false at γ_0). Let $G''_{(u'',v'')}$ be the $\prec$-minimal isomorphic image of $G'_{(u,v)}$. By construction, $G''_{(u'',v'')} \lesssim G_{(u,v)}$. We know three things about $G''_{(u'',v'')}$: (1) Because P is topology based, and, by construction, $P(u,v,G',\gamma_0)$, we thus have $P(u'',v'',G'',\gamma_0)$. (2) $G''_{(u'',v'')} \in \mathcal{B}''$ by construction. (3) $G''_{(u'',v'')} \in \mathcal{B}'$ because there is no proper birooted subgraph of $G''_{(u'',v'')}$ that satisfies P at γ_0 (otherwise, by P being topology based, this contradicts the minimality of $G'_{(u,v)}$). Therefore, $G''_{(u'',v'')} \in \mathcal{B}$. Consequently, $P^+_{\mathcal{B}}(u,v,G,\gamma)$.

Conversely, suppose $P^+_{\mathcal{B}}(u,v,G,\gamma)$. By definition, there exists birooted graphs $G'_{(u,v)}$ and $G''_{(u'',v'')}$ such that $P(u'',v'',G'',\gamma_0)$ and $G''_{(u'',v'')} \cong G'_{(u,v)} \subseteq G_{(u,v)}$. Because P is topology based, we have $P(u,v,G',\gamma)$. By the monotonicity of P, we have $P(u,v,G,\gamma)$.

We now turn to the second statement of the theorem.

Suppose we are given a topology-based and monotonic policy predicate P, and a set $\mathcal{B}$ constructed from P as above. We begin with the only-if direction ($\Rightarrow$). Suppose P is local. Consider $G_{(u,v)} \in \mathcal{B}$ and $e \in E(G)$. By the construction of $\mathcal{B}$, removing e from G will render P false. Because P is local, it means that e is connected to both u and v, as required.

We then turn to the if direction ($\Leftarrow$) of the second statement. Say a set of birooted graphs is ***proper*** whenever every edge of each graph is connected to the two roots. Suppose it is impossible to find a proper set $\mathcal{B}'$ for which $P = P^+_{\mathcal{B}'}$. Then $\mathcal{B}$, as constructed above, is not proper. There is now a birooted graph $G_{(u,v)} \in \mathcal{B}$ and an edge $e \in E(G)$ for which e is not connected to both u and v. By the construction of $\mathcal{B}$, $\neg P(u,v,G-e,\gamma_0)$ but $P(u,v,G,\gamma_0)$. Thus P is not local. □

The above theorem fully characterizes a monotonic relational policy by a set $\mathcal{B}$ of birooted graphs in which edges are always connected to roots. Intuitively, $\mathcal{B}$ specifies relational "patterns" between the accessor and the owner that must be present in the social graph in order for access to be granted. Such a characterization is instrumental to reasoning about monotonic relational policies: it allows us to reason about such policies in graph theoretic terms. A common thinking pattern that has emerged in our ongoing, related work is the following: if a monotonic relational policy $P^+_{\mathcal{B}}$ is satisfied by an owner u, an accessor v and a social graph G, then there must be a subgraph G' of G for which $G'_{(u,v)}$ is isomorphic to a member of $\mathcal{B}$. Now the graph properties shared by members of $\mathcal{B}$ can be applied to $G'_{(u,v)}$. For example, one such property is that all edges in G' are connected to u and v, but there may be other graph-theoretic properties induced by a specific choice of $\mathcal{B}$.

In a symmetric manner, one can establish the following characterization regarding anti-monotonic relational policies:

Corollary 10. *A policy predicate P is both topology based and anti-monotonic iff it is negatively induced by a set $\mathcal{B}$ of birooted graphs. Moreover, P is local iff every edge of each birooted graph in $\mathcal{B}$ is connected to the two roots.*

Again, such a characterization grants us a means to reason about anti-monotonic relational policies in graph theoretic terms.

6 A Case Study: E-learning

SNSs can serve as a generic infrastructure for information sharing beyond recreational purposes [33, 20]. We demonstrate here the utility of general policy predicates in facilitating controlled dissemination of information in a hypothetical information sharing system. An e-learning system [5] performs a variety of tasks related to learning, such as supporting different learning scenarios (e.g. self-study or guided learning), authoring and delivery of learning objects, tutoring, communication, performance evaluation, annotation, administration, etc. Embedded with tools for blogging, podcasting, or social book-marking, today's e-learning environments support social learning [43]. Furthermore, a personal portfolio tool, namely e-portfolio [41], has become a part of e-learning to allow learners to create and showcase their own work (e.g., learning records, artifacts, etc.), in a manner similar to an SNS user profile. Consider a hypothetical e-learning environment modeled as a SNS, adopting the access control model articulated in Sect. 3. We examine how general policy predicates can naturally cater to various access control needs of actors in such an e-learning environment.

Peer help. Peer help is a pervasive phenomenon in learning environments. Suppose peer help is modeled as a profile item of the helper, and help-seekers access that profile item in order to obtain assistance. A learner can only afford to help so many of her peers. Using $\mathsf{distance}_k$ as an access policy for the peer help profile item, a learner can restrict her service only to users within a manageable social circle.

Review. For fairness and privacy, a blind review is an effective peer-reviewing process. When an e-learner wants to try out her seminal ideas, she may prefer to make her ideas accessible only to someone at "arm's length", thereby soliciting feedback outside of her circle of close neighbors. Suppose that the new idea is posted as a profile item. Adopting the anti-monotonic policy $\neg\mathsf{clique}_k$ as the access policy of that item facilitates "arm's length" review.

Initiation. When a learner joins a new learning community (e.g., a class), common friends can play the role of introducer between two strangers. A learner may choose to consider someone to be a potential friend only if they share at least k common friends. Each of the common friends can be viewed as a vote of confidence towards the reputation of a person. This can be arranged by imposing $\textsf{common-friends}_k$ as the communication policy for the friendship invitation primitive.

Meeting places. Recall that a liberal search policy (e.g., $\textsf{everyone}$) destroys the capability nature of user search listings. Yet, search listings need to be reachable before a new user can even start accumulating friends. How does one bootstrap friendship articulation without completely compromising the capability nature of search listings? An idiom is to exploit interest groups as "meeting places". Recall that interest groups can be modeled as virtual users, and group membership can be modeled by being adjacent to the virtual user (Sect. 4.1). The SNS can set up its search policy space to contain only policies of the form $\textsf{common-friends}_{k,V}$, where V is the set of virtual users representing interest groups. In that way, a user becomes reachable through global name search only if the accessor shares k interests with her. In essence, a new user can "meet people" (become searchable) by joining interest groups.

Exclusivity. Learning objects are grouped into modules to provide a manageable coverage of the course materials. One can create virtual users for representing learning modules. Befriending such a virtual user corresponds to subscribing to the underlying module. It is sometimes necessary to ensure that learners are only subscribed to at most k modules. To achieve this effect, the communication policy of the friendship invitation primitive of the virtual users can be set to $\textsf{bad-company}_{k,V}$, where V is the set of virtual users corresponding to the learning modules. Consequently, befriending (i.e., subscription) will be closed for learners who have already subscribed to k modules.

Continuing with the theme of exclusivity, suppose that an essay competition is being organized within a class. Half of the class has been selected to enter the competition. The competing entries will be "graded" by fellow classmates. To avoid conflict of interests, only the non-competing learners in the class are allowed to access the gradable essays posted on the profiles of the competing learners. To model this, we use a virtual user $\textsf{cls}$ to represent the class, and another virtual user $\textsf{com}$ to represent the competition. Befriending $\textsf{cls}$ represents class membership, and, likewise, befriending $\textsf{com}$ represents entering the competition. Now, the access policies of the competing essays can be set to $\textsf{bad-company}_{0,\{\textsf{cls}\}} \wedge \textsf{common-friends}_{1,\{\textsf{com}\}}$.

7 An Executable Encoding

We developed an executable encoding of the access control model presented in Sect. 3. Specifically, we developed a *framework* of Prolog predicates that capture the dynamic behavior of the access control model. This implementation exercise is conducted with two purposes in mind. Firstly, the executable encoding allows researchers to "try out" the mathematical model as well as various instantiations in order to gain a deeper understanding of how the model behaves and what it is capable of. Secondly, the executable encoding can be used as a policy enforcement engine in an actual Facebook-style SNS.

With the two purposes in mind, we want our executable encoding to possess two characteristics:

1. The encoding shall retain the declarative nature of the model as defined in Sect. 3. We desire a close parallel between the various theoretical entities (e.g., the inference rules of Figs. 2, 3 and 5) and their executable encoding.

2. The encoding shall be customizable, in the sense that it should be easy for programmers to instantiate the model by specific choices of policy spaces and communication automata.

To address the first requirement, we used the Prolog family of declarative languages for our executable encoding. We began our development exercise with SWI-Prolog [2], a Prolog compiler in the Edinburgh Prolog family. We ran into an issue under this development environment. A direct encoding of the recursive structure of the inference rules (Figs. 2, 3 and 5) caused non-termination with the standard Prolog backtracking mechanism. However, tweaking the clauses to avoid non-termination rendered the resulting clauses highly procedural, and destroys the parallel between the executable encoding and the original specification. We ended up adopting B-Prolog [1] as our specification language. B-Prolog offers a tabling facility, which preserves most of the recursive structures in our specification while guaranteeing termination. As a result, our final encoding closely resembles the original paper specification.

To address the second requirement (customizability), we developed an extensible *framework* of Prolog clauses, in which it is possible for programmers to further instantiate the framework through the specification of, for example, subject and object sets, communication automata, adjacency predicates, and policy spaces for their specific SNSs. We also devised an encoding of sets (e.g., subject sets, sets of communication primitives, etc) and functions (e.g., transition functions of communication automata, policy predicates, policy assignments, etc) that is on the one hand highly efficient in the presence of indexing, and on the other hand supportive of customizability. As an exercise, we instantiated this framework to encode $\mathcal{FB}_{lite}$, as well as various policy predicates mentioned in the previous sections.

Some statistics regarding the implementation effort are provided below:

Component	**# of comment-free lines of code**	**# of clauses**
Core framework	94	24
$\mathcal{FB}_{lite}$ instantiation	91	76

8 Related Work

For general studies on the phenomenon of social networks, consult the recent special issue of the *Journal of Computer-Mediated Communication* on Social Network Sites. The editorial article of boyd and Ellison contains a survey of privacy and security issues in Social Network Systems [17]. An early work on social network privacy is [24], which highlights potential privacy attacks on social networks. [3] studies the relationship between user demographics and privacy settings in social networks. Phishing attacks on social networks are discussed in [27]. The privacy impact of the News Feed feature of Facebook is studied in [16]. The lack of flexibility of existing privacy settings in social networks is analyzed in [35].

To the best of our knowledge, this is the first work to provide a formal articulation of the access control paradigm behind the Facebook privacy preservation mechanism. We argue in Sect. 2.2 that the access control paradigm behind Facebook is distinct from capability systems [18, 32], Discretionary Access Control (DAC) [23, 30] and Role-Based Access Control (RBAC) [37, 21]. We also compared this access control paradigm to history-based access control [39] by identifying the history information consumed by the authorization mechanism. Consequently, our work is related to [22]. While both [22] and this work employ the idea of abstraction to model information loss, in this work we attempt to characterize the information that is actually used in making authorization decisions, rather than the information monitored by the authorization mechanisms. A comparison with TMSs [9, 42] can also be found in Sect. 2.2.

Perhaps closest in spirit to our methodology is that of Weeks [42], who proposes a formal framework for delineating the design space of Trust Management Systems (TMSs). A concrete TMS is obtained by instantiating the framework with a concrete lattice of authorization labels and a concrete license vocabulary. Each license is specified as a higher-order function via the lambda notation. The meaning of authorization is specified by a fixed-point semantics. The model has been instantiated to simulate the TMSs KeyNote and SPKI. Our work is similar in that our SNS model is parameterized by a vocabulary of policies (specified as lambda expressions) and a consent protocol (specified as a communication automaton and an adjacency predicate). Our approach defers from that of Weeks in that we specify the semantics of authorization by way of an operational semantics (i.e., an abstract state machine).

A number of proposals, in various level of maturity, attempt to advance beyond the access control mechanisms found in commercial SNSs. To promote the usability of access control in social computing, Hart *et al.* propose to automatically infer default access control policies based on the contents of user data [26]. To preserve the trustworthiness of user constructed data in SNSs, Ali *et al.* propose to use trust metrics to impose access restrictions akin to multi-level security [4]. Kruk *et al.* considers the combination of asymmetric friendship, trust metrics and degree-of-separation policies (i.e., $\mathsf{distance}_k$) in a distributed identity management system based on social networks [28]. The most mature of these proposals is that of Carminati *et al.*, in which a decentralized social network system with relationship types, distributed trust metrics and degree-of-separation policies is developed [12, 13, 10, 14]. Our model assumes a fully mediated environment, as opposed to Kruk *et al.* and Carminati *et al.*, and thus enjoys the richness offered by Stage-I authorization (i.e., search and traversal policies, search listings as capabilities, etc). Although our model does not support asymmetric friendship, friendship types and trust metrics, it supports such socially interesting policies as $\mathsf{common\text{-}friends}_k$ and clique_k, as well as anti-monotonic policies for privacy preservation.

Another relevant line of research addresses the need of policy languages for expressing access control policies unique to SNSs. Of particular interest is the work of Carminati *et al.* [11], who adopt semantic web technologies, including the Resource Description Framework (RDF) and the Web Ontology Language (OWL) to describe user profiles, relationships among users, resources (i.e, objects), relationship between users and resources, and actions (i.e., communication primitives). This 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. In our model, the characterization theorem implies that relational policies can be expressed as sets of birooted graphs. Essentially, birooted graphs could be seen as a visual means for specifying relational policies in Facebook-style SNSs.

PriMa [40] 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 graph 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 mechanism by which access control policies are automatically constructed for users. The mechanism consists of three major features. First, the default privacy preference for a profile item is inferred from the average privacy preference of similar and related users (which is called the crowd in [40]). Second, the privacy preference of a profile item is then employed to assess its sensitivity. Specifically, the sensitivity of a profile item is proportional to its privacy preference (i.e., the more private, the more sensitive), and how popular the owner is (i.e., the more connected is the owner, the more sensitive its profile items), but inversely proportional to the accessibility of the same profile item of the crowd by the owner (i.e., if my peers do not grant me access, then I better do not grant access easily). Third, access control rules are then generated from the sensitivity values. In particular, accessibility is proportional to the closeness of the owner and the accessor, and inversely proportional to the risk of disclosure. Closeness is proportional to the number of common friends shared by the owner and the accessor. Risk is proportional to the sensitivity of the profile item and the popularity of the accessor. Note that the scheme above lays bare the strategic role played by topology-based concepts such as common friends and popularity.

An aspect of access control not addressed by our work is the protection of users from third-party applications. Facebook provides a programming platform through which a third-party developer may construct software components that augment the capability of Facebook. (OpenSocial is a similar Application Programming Interface (API) developed by Google.) These third-party applications often access the personal information of a user and her friends in order to deliver their functionalities. Yet the applications are actually hosted on a server over which Facebook has no control. This motivates the development of access control to prevent malicious third-party applications from misusing the information they access through the Facebook API. Besmer *et al.* [8] proposed a fine-grained access control model for limiting the capability of third-party applications. In their model, not only can a profile owner impose access restrictions on applications, but a security-conscious accessor can also set up policies to protect owners of profiles accessed by a third-party application on behalf of the accessor. Besmer *et al.* also evaluated the usability of this scheme.

9 Future Work and Open Questions

This work is but the first step of the three-pronged research agenda articulated in Sect. 1. We plan to address challenge (b), by identifying security properties that should be enforced in instantiations of our SNS model. Specifically, we are particularly concerned about the possibility of Sybil Attacks in the presence of relational policies. Consider the monotonic relational policy $\mathsf{celebrity}_k \wedge \mathsf{distance}_l$. A malicious accessor within distance l from the owner could create a large number ($> k$) of fake friends and then gain access. Such attacks are also possible with careless uses of anti-monotonic policies. We are working on the formal characterization of SNS instantiations that would render such attacks provably impossible.

We are also interested in addressing challenge (c), by the design of visualization tools to help users anticipate the privacy implications of their actions. Our early work on a visualization technique known as Reflective Policy Assessment (RPA) can be found in [6].

Another direction is to further generalize the model to account for richer forms of acquaintance relations and policies, including relationship types and asymmetric acquaintance. Relational policies in such a model will then be articulated in terms of not only simple graph edges, but also the specific relationship types each edge represents, and the direction of such a relationship. It is conjectured that, with the enriched model, one can apply SNSs as the basis of a complex information sharing system. Could this type of access control systems be a natural match for the protection needs found in many professional domains, including the medical one, in which trust is a function of the professional relationships between owners and accessors?

An implication of the characterization results in Sect. 5 is that birooted graphs can be used as a policy language for expressing monotonic and anti-monotonic relational policies in the model presented in this work. Moreover, such a language is provably *adequate* in that every monotonic or anti-monotonic relational policy can be expressed this way. Extending our model to incorporate relationship types and asymmetric acquaintances gives rise to a analogous research question: What is an *adequate* policy language for expressing relational policies in the extended model?

10 Conclusions

We have formalized the distinct access control paradigm behind the Facebook privacy preservation mechanism into an access control model, which delineates the design space of protection mechanisms under this paradigm of access control. We have also demonstrated how the model can be instantiated to express access control policies that possess rich and natural social significance. We formally characterized the family of relational policies, which are a distinct feature of Facebook-style SNSs. A executable encoding of of our model has also been developed.

Acknowledgments. This work is supported in part by an NSERC Strategic Project Grant. We thank Howard Hamilton for introducing us to Facebook-style SNSs. We thank Jason Crampton and Ida Siahaan for their constructive feedback on earlier manuscripts of this work.

References

[1] B-Prolog. `http://www.probp.com`.

[2] SWI Prolog. `http://www.swi-prolog.org`.

[3] Alessandro Acquisti and Ralph Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *Proceedings of the 6th International Workshop on Privacy Enhancing Technologies (PET'06)*, volume 4258 of *Lecture Notes in Computer Science*, pages 36–58, Cambridge, UK, June 2006. Springer.

[4] Bader Ali, Wilfred Villegas, and Muthucumaru Maheswaran. A trust based approach for protecting user data in social networks. In *Proceedings of the 2007 Conference of the Center for Advanced Studies in Collaborative Research (CASCON'07)*, pages 288–293, Richmond Hill, Ontario, Canada, October 2007.

[5] Mohd Anwar. Identity and reputation management for online learners. In *Proceedings of the 2008 International Conference on Intelligent Tutoring Systems (ITS'08) – Young Researcher's Track*, pages 177–187, Montreal, Canada, June 2008.

[6] Mohd Anwar, Philip W. L. Fong, Xue-Dong Yang, and Howard Hamilton. Visualizing privacy implications of access control policies in social network systems. Technical Report 2009-927-06, University of Calgary, May 2009.

[7] Ezedin S. Barka and Ravi S. Sandhu. Framework for role-based delegation models. In *Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC'00)*, New Orleans, Louisiana, USA, December 2000.

[8] Andrew Besmer, Heather Richter Lipford, Mohamed Shehab, and Gorrell Cheek. Social applications: Exploring a more secure framework. In *Proceedings of the 5th Symposium on Usable Privacy and Security (SOUPS'09)*, Mountain View, California, USA, July 2009.

[9] 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.

[10] 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.

[11] Barbara Carminati, Elena Ferrari, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. 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.

[12] 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.

[13] 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.

[14] 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.

[15] Jason Crampton and Hemanth Khambhammettu. Delegation in role-based access control. *International Journal of Information Security*, 7(2):123–136, April 2008.

[16] danah boyd. Facebook's privacy trainwreck: Exposure, invasion, and social convergence. *Convergence: The International Journal of Research into New Media Technologies*, 14(1):13–20, February 2008.

[17] danah m. boyd and Nicole B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, October 2008.

[18] Jack B. Dennis and Earl C. Van Horn. Programming semantics for multiprogrammed computations. *Communications of the ACM*, 9(3):143–155, March 1966.

[19] Reinhard Diestel. *Graph Theory*. Springer, 3rd edition edition, 2006.

[20] Joan Dimicco, David R. Millen, Werner Geyer, Casey Dugan, Beth Brownholtz, and Michael Muller. Motivations for social networking at work. In *Proceedings of the ACM 2008 Conference on Computer Supported Cooperative Work (CSCW'08)*, pages 711–720, San Diego, California, USA, November 2008.

[21] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, 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.

[22] Philip W. L. Fong. Access control by tracking shallow execution history. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy (S&P'04)*, pages 43–55, Berkeley, California, USA, May 2004.

[23] 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.

[24] Rolph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society (WPEA'05)*, pages 71–80, Alexandria, VA, USA, November 2005.

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

[26] Michael Hart, Rob Johnson, and Amanda Stent. More content – less control: Access control in the Web 2.0. In *Proceedings of the 2007 Workshop on Web 2.0 Security and Privacy (W2SP'07)*, pages 1–3, Oakland, California, USA, May 2007.

[27] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, October 2007.

[28] 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 *LNCS*, pages 140–154, Beijing, China, September 2006. Springer.

[29] Ninghui Li, John C. Mitchell, and William H. Winsborough. Beyond proof-of-compliance: Security analysis in trust management. *Journal of the ACM*, 52(3):474–514, May 2005.

[30] 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.

[31] R. J. Lipton and L. Snyder. A linear time algorithm for deciding subject security. *Journal of the ACM*, 24(3):455–464, July 1977.

[32] Mark S. Miller, Ka-Ping Yee, and Jonathan Shapiro. Capability myths demolished. Technical Report SRL2003-02, System Research Lab, Department of Computer Science, The John Hopkins University, Baltimore, Maryland, USA, 2003.

[33] Junichiro Mori, Tatsuhiko Sugiyama, and Yutaka Matsuo. Real-world oriented information sharing using social networks. In *Proceedings of the 2005 ACM SIGGROUP Conference on Supporting Group Work (GROUP'05)*, pages 81–84, Sanibel Island, Florida, USA, November 2005.

[34] Benjamin C. Pierce. *Types and Programming Languages*. MIT Press, 2002.

[35] Sören Preibusch, Bettina Hoser, Seda Gürses, and Bettina Berendt. Ubiquitous social networks - opportunities and challenges for privacy-aware user. In *Proceedings of the Workshop on Data Mining for User Modeling*, pages 50–62, Corfu, Greece, June 2007.

[36] Zick Rubin. Disclosing oneself to a stranger: Reciprocity and its limits. *Journal of Experimental Social Psychology*, 11(3):233–260, May 1975.

[37] 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.

[38] Ravinderpal Singh Sandhu. The schematic protection model: Its definition and analysis for acyclic attenuating schemes. *Journal of the ACM*, 35(2):404–432, April 1988.

[39] Fred B. Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1):30–50, February 2000.

[40] 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.

[41] David Tosh, Tracy Penny Light, Kele Fleming, and Jeff Haywood. Engagement with electronic portfolios: Challenges from the student perspective. *Canadian Journal of Learning and Technology*, 31(3), Fall 2005.

[42] 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.

[43] Etienne Wenger. Communities of practice and social learning systems. *Organization*, 7(2):225–246, 2000.