

A Privacy Preservation Model for Facebook-Style Social Network Systems

Philip W.L. Fong¹, Mohd Anwar¹, and Zhen Zhao²

¹ Department of Computer Science, University of Calgary, Alberta, Canada
{pwl.fong, manwar}@ucalgary.ca

² Department of Computer Science, University of Regina, Saskatchewan, Canada
zhao112z@uregina.ca

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 privacy preservation 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 but possess rich and natural social significance. This work thus delineates the design space of privacy preservation mechanisms for Facebook-style social network systems, and lays out a formal framework for policy analysis in these 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 [1], 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 [2,3] in the midst of discretionary access control [4,5]. 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 [6,7,8], or, more recently, security analysis [9,5].

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 threefold:

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 [10] of the global protection state, namely, the social graph.
2. We formalized the above insight into a concrete access control model for delimiting the design 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 topology-based access control policies that possess rich and natural social significance: e.g., degree of separation, known quantity, clique, trusted referral, 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.

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 case study of modeling an e-learning system as an instantiation of our access control model is provided in Sect. 5. Sect. 6 surveys related literature. Conclusions and future work are given in Sect. 7.

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.

Search Listings and their Reachability. Access to profile items is authorized in two stages. 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” [11,12] of the user in the system, through which access is mediated. There are two means by which a profile 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 omit¹. Not every accessing user sees the same profile items when a profile

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

is displayed. The owner may assign 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. 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 systems [4,5], 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 [12]. 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 and networks 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.* The 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 [11,12]. 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 [12].

D2 *Relation-Based Policies.* Due to the lack of a global name space for accessible resources (a common feature in capability systems [12]), privileged users are not specified in policies by names. Instead, they are specified *intensionally*² as the set

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., $S = \{x \in \mathbb{N} \mid x < 3\}$).

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 [4,5], nor to a centrally administrated set of roles, as in the case of RBAC [13,14]. Instead, privileges are granted with respect to an intentionally-specified relation, the articulation of which is carried out in a distributed manner.

D3 *Abstraction of Communication History.* As in many access control systems [15], 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 [10], into a social graph, the topology of which becomes a basis of authorization decisions.

Perhaps the access control paradigm that is the most comparable to that of Facebook is Trust Management Systems (TMSs) [16,17]. To fix thoughts, we provide a comparison with the family of TMSs identified by Weeks [17]. 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 (**D2**). Second, Facebook is completely mediated, and thus search listing reachability (Stage I) is a precondition of authorization (**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 (**D3**). In our generalization below, this allows us to formulate topology-based policies that have no analogue in Weeks' TMSs.

Generalization. Facebook embodies the above paradigm of access control (**D1–D3**) by providing:

G1 a specific protocol for establishing acquaintance, and

G2 a specific family of relation-based 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 **D1–D3**, while allowing an arbitrary consenting mechanism (**G1**) and policy vocabulary (**G2**) to be adopted. Therefore, such a model delineates the design space of access control mechanisms embodying such a paradigm.

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 , $\mathcal{P}_k(S)$ is the set of all size- k subsets of S , and, when S

is finite, $\mathcal{G}(S)$ is the set of all simple graphs with S as the vertex set (i.e., $\mathcal{G}(S) = \{ \langle S, E \rangle \mid E \subseteq \mathcal{P}_2(S) \}$). We use the standard λ -notation for constructing functions [18]: 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 $S \rightarrow T$ for the set of all partial functions with a subset of S as the domain and T as the codomain. Given $f \in S \rightarrow T$, $s \in S$, and $t \in T$, we write $f[s \mapsto t]$ to denote the function $(\lambda x . \text{if } x = s \text{ then } t \text{ else } f(x))$.

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.

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*.

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 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 pair of users $\{u, v\}$, we define an identification function $\iota_{\{u,v\}} : \{u, v\} \rightarrow \mathbb{B}$ 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 pair. 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 [15] to model the communication protocol between user pairs, as well as to track communication history. We reuse the notational convention in [10]. 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 **start state**, and $\delta : \Gamma \times \mathbb{B} \times \Sigma \rightarrow \Gamma$ 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 communication protocol.

As we shall see in the next section, a SNS tracks, at run time, a mapping $His : \mathcal{P}_2(Sub) \rightarrow \Gamma$, 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.

Acquaintance Topology. The communication state between a pair of users is *local* in nature, describing only the communication history between a 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**, $Adj : \Gamma \rightarrow \mathbb{B}$, 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:

$$SG(Adj, His) = \lambda(Adj, His) . \langle Sub, \{\{u, v\} \in \mathcal{P}_2(Sub) \mid Adj(His(\{u, v\}))\} \rangle$$

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.

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 $Sub \times Sub \times \mathcal{G}(Sub) \times \Gamma \rightarrow \mathbb{B}$. 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.

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 $Sub \times Sub \times \mathcal{G}(Sub) \times \Gamma \rightarrow \mathbb{B}$), 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.

System. A Facebook-style SNS, or a **system** in short, is an pentuple $N = \langle Sub, Obj, M, Adj, PS \rangle$. 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 $Sub \times Obj$. $M = \langle \Sigma, \Gamma, \gamma_0, \delta \rangle$ is a CA. $Adj : \Gamma \rightarrow \mathbb{B}$ is an adjacency predicate. $PS = \{PS_r\}_{r \in \mathcal{R}_N}$ is a family of **policy spaces** indexed by **resources** $r \in \mathcal{R}_N$, such that $\mathcal{R}_N = \{\text{search, traversal}\} \cup \Sigma \cup Obj$, and each PS_r is a countable set of policy predicates (i.e., with type signature $Sub \times Sub \times \mathcal{G}(Sub) \times \Gamma \rightarrow \mathbb{B}$). Intuitively, PS_{search} specifies the set of policy predicates that users may legitimately adopt as their search policies, while $PS_{\text{traversal}}$, 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 $o \in Obj$. 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.

$$\begin{array}{c}
S \vdash_N u \text{ finds } u \quad (\text{F-SLF}) \\
\hline
\frac{N = \langle _, _, _, Adj, _ \rangle \quad G = \text{SG}(Adj, His) \quad \{u, v\} \in E(G)}{\langle His, Pol \rangle \vdash_N v \text{ finds } u} \quad (\text{F-FRD}) \\
\hline
\frac{N = \langle _, _, M, Adj, _ \rangle \quad M = \langle _, _, \gamma_0, _ \rangle \quad \gamma = His_{\langle \gamma_0 \rangle}(\{u', v\})}{G = \text{SG}(Adj, His) \quad \{u, u'\} \in E(G) \quad Pol(u', \text{traversal})(u', v, G, \gamma)} \quad (\text{F-TRV}) \\
\hline
\langle His, Pol \rangle \vdash_N v \text{ finds } u \\
\hline
\frac{N = \langle _, _, M, Adj, _ \rangle \quad M = \langle _, _, \gamma_0, _ \rangle \quad \gamma = His_{\langle \gamma_0 \rangle}(\{u, v\})}{G = \text{SG}(Adj, His) \quad Pol(u, \text{search})(u, v, G, \gamma)} \quad (\text{F-SCH}) \\
\hline
\langle His, Pol \rangle \vdash_N v \text{ finds } u
\end{array}$$

Fig. 1. Definition of the reachability sequent $S \vdash_N v \text{ finds } u$

3.2 System States

State. Suppose a system $N = \langle Sub, Obj, M, Adj, PS \rangle$ 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 $S = \langle His, Pol \rangle$:

- $His : \mathcal{P}_2(Sub) \rightarrow \Gamma$ maps each pair of users to their current communication state. Given $\gamma \in \Gamma$, we also define $His_{\langle \gamma \rangle} : \mathcal{P}_2(Sub) \cup \mathcal{P}_1(Sub) \rightarrow \Gamma$ to be the function $(\lambda\{u, v\}. \text{if } u = v \text{ then } \gamma \text{ else } His(\{u, v\}))$. That is, $His_{\langle \gamma \rangle}$ is the extension of His that maps $\{u, v\}$ to γ whenever $u = v$.
- $Pol : Sub \times \mathcal{R} \rightarrow \bigcup_{r \in \mathcal{R}} PS_r$ is a mapping that records the current policy for every resource of every user. It is required that $\forall u \in Sub. \forall r \in \mathcal{R}. Pol(u, r) \in PS_r$.

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.

Reachability. Fig. 1 describes the rules for navigating the social graph. Specifically, the sequent “ $S \vdash_N v \text{ finds } u$ ” holds whenever accessor v is permitted to traverse the social graph to reach the search listing of user u . According to Fig. 1, this occurs if $v = u$ (F-SLF), if v is adjacent to u in the social graph (F-FRD), if v may reach a neighbor u' of u , and the traversal policy of u' allows v to access the friend list of u' (F-TRV), or, lastly, if 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.

Accessibility. Fig. 2 specifies the rules for object access. Specifically, the sequent “ $S \vdash_N v \text{ reads } u.o$ ” holds whenever accessor v is permitted to access object o of owner u . According to Fig. 2, 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

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

Fig. 2. Definition of the accessibility sequent $S \vdash_N v \text{ reads } u.o$

$$\begin{array}{l}
 \mathcal{T}_N \ni t ::= \text{com}(v, u, a) \text{ for } u, v \in \text{Sub}, a \in \Sigma \\
 \quad | \text{pol}(u, r, P) \text{ for } u \in \text{Sub}, r \in \mathcal{R}_N, P \in PS_r
 \end{array}$$

Fig. 3. Definition of the set \mathcal{T}_N of transition identifiers for a system $N = \langle \text{Sub}, \text{Obj}, M, Adj, PS \rangle$, where $M = \langle \Sigma, \Gamma, \gamma_0, \delta \rangle$

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

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

Fig. 4. Definition of the state transition relation $S \xrightarrow{t}_N S'$

Fig. 3. The convention is that the first argument of a constructor is always the initiator of the transition. We write $\text{initiator}(t)$ for the initiator of transition identifier t .

Fig. 4 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 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 .

3.4 Monotonicity, Propriety and Definability

A policy predicate P is said to be **monotonic** iff $P(u, v, G, \gamma) \Rightarrow P(u, v, G + e, \gamma)$ for every $u, v \in \text{Sub}$, $G \in \mathcal{G}(\text{Sub})$, $e \in \mathcal{P}_2(\text{Sub})$, and $\gamma \in \Gamma$. Here, $G + e$ denotes the graph obtained by adding an extra edge e into graph G . 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. Conversely, a policy predicate P is said to be *anti-monotonic* iff $P(u, v, G + e, \gamma) \Rightarrow P(u, v, G, \gamma)$ for every $u, v \in Sub, G \in \mathcal{G}(Sub), e \in \mathcal{P}_2(Sub)$, and $\gamma \in \Gamma$. 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. Note that both monotonicity and anti-monotonicity are preserved by the policy combinators \wedge and \vee . As expected, $\neg P$ is anti-monotonic if P is monotonic, and vice versa.

A state S_0 is a *proper initial state* whenever the following conditions are met:

1. The communication state between every pair of users is γ_0 .
2. The sequent $S_0 \vdash_N v$ finds $u.o$ is false whenever $u \neq v$. (Consequently, $S_0 \vdash_N v$ reads $u.o$ is false whenever $u \neq v$. That is, a search listing is reachable only from its owner, and thus Stage-I authorization fails uniformly in such a state.)

This notion of propriety gives us a manageable fixed point for policy analysis in future work. A system has proper initial states iff it satisfies the following conditions:

- $Adj(\gamma_0) = 0$. (Consequently, F-FRD is rendered inapplicable.)
- PS_{search} contains a predicate that returns 0 when the social graph has no edge or when the communication state is γ_0 . (Thus, F-SCH can be rendered inapplicable.)

A system that satisfies these two conditions is *well-formed*. Well-formed systems have proper initial states. From now on we consider only well-formed systems.

A state S is *definable* iff it is reachable from some proper initial state S_0 (i.e., $S_0 \xrightarrow{w} S$ for some $w \in (\mathcal{T}_N)^*$). We consider only definable states in the sequel. Given a concrete system, a natural task is to characterize the set of all definable states.

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

The communication automaton $M = \langle \Sigma, \Gamma, \gamma_0, \delta \rangle$ is defined such that $\Sigma = \{ \text{invite, accept, ignore, remove} \}$, $\Gamma = \{ \text{stranger, invited-1, invited-0, friend} \}$, $\gamma_0 = \text{stranger}$, and δ is defined as in Fig. 5.

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

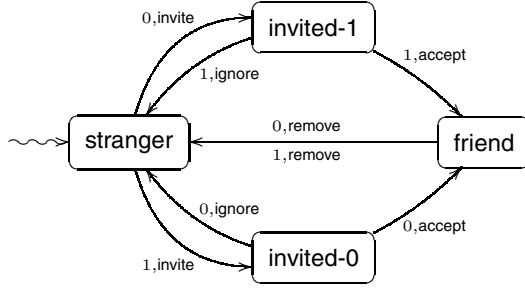


Fig. 5. Transition diagram for the communication automaton of \mathcal{FB}_{lite}

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

Fig. 6. A list of Facebook-inspired policy predicates

The traversal policy space is $PS_{traversal} = \{\text{no-one, only-me, only-friends, friends-of-friends, everyone}\}$, where the policy predicates are defined in Fig. 6.

The search policy space PS_{search} could have been defined in the same way as $PS_{traversal}$ 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:

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

This predicate returns true iff u has extended a friendship invitation to v . Then PS_{search} is defined as $\{P \vee \text{owner-invited} \mid P \in PS_{traversal}\}$. 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 $PS_{traversal}$. 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 $PS_{\text{Basic-Information}} = PS_{search}$.

The communication policy space is defined as follows:

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

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 $PS_a = \{\text{everyone}\}$ for most a . Secondly, a user may not always want to allow friendship invitations from strangers. PS_{invite} is therefore set to $\{\text{no-one, friends-of-friends, everyone}\}$.

Proposition 1. $\mathcal{FB}_{\text{lite}}$ is well-formed, with PS containing only monotonic policies. In addition, every state is definable.

$\mathcal{FB}_{\text{lite}}$ does not capture all aspects of the access control mechanism of Facebook (see [19, Sect. 4.1] for a list of missing features). Nevertheless $\mathcal{FB}_{\text{lite}}$ 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. A policy similar to friends-of-friends will allow group members to access objects owned by one another.

4.2 Topology-Based Policies

This section explores policies other than those already offered by Facebook. The goal is to illustrate the possibilities supported by the proposed model. Specifically, we consider policies that are based on topological information provided by the social graph (see [19, Sect. 4.3] for an example of policies based on communication history). It is assumed that adjacency in the social graph is induced by some form of social acquaintance (e.g., friendship), which in turn is formed by a mutual consent protocol (e.g., friendship invitation and acceptance). Our focus here is on access policies:

Degree of Separation. For $k \geq 1$, let policy 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 . 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 an 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. Notice also that the communication history γ between u and v is not taken into consideration in authorization, and thus the policy is purely topology-based.

Known Quantity. For $k \geq 1$, let policy common-friends_k be the following predicate:

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

where $N_G(u)$ is the *neighborhood* of u in graph G , which is defined to be the vertex set $\{v \in V(G) \mid \{u, v\} \in E(G)\}$. 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 $\text{common-friends}_1 = \text{distance}_2$.

Clique. For $k \geq 2$, define policy clique_k as follows:

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

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. 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 $\text{clique}_2 = \text{distance}_1$.

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

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

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 $\text{common-friends}_{k,\text{Sub}} = \text{common-friends}_k$.

Stranger. Consider $\neg \text{distance}_k$, the negation of 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”. Unlike other policies presented in this section, $\neg \text{distance}_k$ is anti-monotonic.

5 A Case Study: E-Learning

SNSs can serve as a generic infrastructure for information sharing beyond recreational purposes [20,21]. We demonstrate here the utility of topology-based policies in facilitating controlled dissemination of information in a hypothetical information sharing system. An e-learning system [22] 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 [23]. Furthermore, a personal portfolio tool, namely e-portfolio [24], 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 topology-based policies 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. A learner can only afford to help so many of her peers. Using distance_k as an access policy, a learner can restrict peer help 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. The anti-monotonic policy -clique_k serves this purpose.

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 common-friends_k as the communication policy for the friendship invitation primitive.

Meeting places. Recall that a liberal search policy (e.g., 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. The SNS can set up its search policy space to contain only policies of the form $\text{common-friends}_{k,V}$, 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.

6 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 [1]. There is also a growing body of literature on the anonymization of social networks (e.g., [25,26]).

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 [11,12], Discretionary Access Control (DAC) [4,5] and Role-Based Access Control (RBAC) [13,14]. We also compared this access control paradigm to history-based access control [15] by identifying the history information consumed by the authorization mechanism. Consequently, our work is related to [10]. While both [10] 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 [16,17] can also be found in Sect. 2.2.

Perhaps closest in spirit to our methodology is that of Weeks [17], 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 [27]. 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 [28]. Kruk *et al.* considers the combination of asymmetric friendship, trust metrics and degree-of-separation policies (i.e., distance_k) in a distributed identity management system based on social networks [29]. The most mature of these proposals is that of Carminati *et al.*, in which a decentralized social network system with relationship types, trust metrics and degree-of-separation policies is developed [30]. 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 common-friends_k and clique_k , as well as anti-monotonic policies for privacy preservation.

7 Conclusions and Future Work

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.

This work is but the first step of the three-pronged research agenda articulated in Sect. 1. We plan to address challenge (b), identifying security properties that should be enforced in instantiations of our SNS model, and challenge (c), the design of visualization tools to help users anticipate the privacy implications of their actions [31]. Another direction is to further generalize the model to account for richer forms of acquaintance relations and policies, including relationship types, asymmetric acquaintance, and optionally specified trust metrics (i.e., specification by enumerating examples).

Acknowledgments. This work is supported in part by an NSERC Strategic Project Grant. We thank Howard Hamilton for introducing us to Facebook-style SNSs.

References

1. boyd, d.m., Ellison, N.B.: Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication* 13(1), 210–230 (2008)
2. Barka, E.S., Sandhu, R.S.: Framework for role-based delegation models. In: *Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC 2000)*, New Orleans, Louisiana, USA (December 2000)
3. Crampton, J., Khambhammettu, H.: Delegation in role-based access control. *International Journal of Information Security* 7(2), 123–136 (2008)
4. Graham, G.S., Denning, P.J.: Protection: Principles and practices. In: *Proceedings of the 1972 AFIPS Spring Joint Computer Conference*, Atlantic City, New Jersey, USA, May 1972, vol. 40, pp. 417–429 (1972)
5. Li, N., Tripunitara, M.V.: On safety in discretionary access control. In: *Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P 2005)*, Oakland, California, USA, May 2005, pp. 96–109 (2005)
6. Harrison, M.A., Ruzzo, W.L., Ullman, J.D.: Protection in operating systems. *Communications of the ACM* 19(8), 461–471 (1976)
7. Lipton, R.J., Snyder, L.: A linear time algorithm for deciding subject security. *Journal of the ACM* 24(3), 455–464 (1977)
8. Sandhu, R.S.: The schematic protection model: Its definition and analysis for acyclic attenuating schemes. *Journal of the ACM* 35(2), 404–432 (1988)
9. Li, N., Mitchell, J.C., Winsborough, W.H.: Beyond proof-of-compliance: Security analysis in trust management. *Journal of the ACM* 52(3), 474–514 (2005)
10. Fong, P.W.L.: Access control by tracking shallow execution history. In: *Proceedings of the 2004 IEEE Symposium on Security and Privacy (S&P 2004)*, Berkeley, California, USA, May 2004, pp. 43–55 (2004)
11. Dennis, J.B., Horn, E.C.V.: Programming semantics for multiprogrammed computations. *Communications of the ACM* 9(3), 143–155 (1966)
12. Miller, M.S., Yee, K.P., Shapiro, J.: Capability myths demolished. Technical Report SRL2003-02, System Research Lab, Department of Computer Science, The John Hopkins University, Baltimore, Maryland, USA (2003)
13. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. *IEEE Computer* 19(2), 38–47 (1996)
14. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, R., Chandramouli, R.: Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security* 4(3), 224–274 (2001)
15. Schneider, F.B.: Enforceable security policies. *ACM Transactions on Information and System Security* 3(1), 30–50 (2000)
16. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trust management. In: *Proceedings of the 1996 IEEE Symposium on Security and Privacy (S&P 1996)*, Oakland, California, USA, May 1996, pp. 164–173 (1996)
17. Weeks, S.: Understanding trust management systems. In: *Proceedings of the 2001 IEEE Symposium on Security and Privacy (S&P 2001)*, Oakland, California, USA, May 2001, pp. 94–105 (2001)
18. Pierce, B.C.: *Types and Programming Languages*. MIT Press, Cambridge (2002)
19. Fong, P.W.L., Anwar, M., Zhao, Z.: A privacy preservation model for Facebook-style social network systems. Technical Report 2009-926-05, University of Calgary (April 2009)
20. Mori, J., Sugiyama, T., Matsuo, Y.: Real-world oriented information sharing using social networks. In: *Proceedings of the 2005 ACM SIGGROUP Conference on Supporting Group Work (GROUP 2005)*, Sanibel Island, Florida, USA, November 2005, pp. 81–84 (2005)

21. Dimicco, J., Millen, D.R., Geyer, W., Dugan, C., Brownholtz, B., Muller, M.: Motivations for social networking at work. In: Proceedings of the ACM 2008 Conference on Computer Supported Cooperative Work (CSCW 2008), San Diego, California, USA, November 2008, pp. 711–720 (2008)
22. Anwar, M.: Identity and reputation management for online learners. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 177–187. Springer, Heidelberg (2008)
23. Wenger, E.: Communities of practice and social learning systems. *Organization* 7(2), 225–246 (2000)
24. Tosh, D., Light, T.P., Fleming, K., Haywood, J.: Engagement with electronic portfolios: Challenges from the student perspective. *Canadian Journal of Learning and Technology* 31(3) (Fall 2005)
25. Thompson, B., Yao, D.: The union-split algorithm and cluster-based anonymization of social networks. In: Proceedings of the 4th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2009), Sydney, Australia, March 2009, pp. 218–227 (2009)
26. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: Proceedings of the 2009 IEEE Symposium on Security and Privacy (S&P 2009), Oakland, California, USA (May 2009)
27. Hart, M., Johnson, R., Stent, A.: 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 2007), Oakland, California, USA, May 2007, pp. 1–3 (2007)
28. Ali, B., Villegas, W., Maheswaran, M.: 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 2007), Richmond Hill, Ontario, Canada, October 2007, pp. 288–293 (2007)
29. Kruk, S.R., Grzonkowski, S., Gzella, A., Woroniecki, T., Choi, H.-C.: D-FOAF: Distributed identity management with access rights delegation. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 140–154. Springer, Heidelberg (2006)
30. Carminati, B., Ferrari, E., Perego, A.: Enforcing access control in web-based social networks. *ACM Transactions on Information and System Security* (to appear, 2009)
31. Anwar, M., Fong, P.W.L., Yang, X.D., Hamilton, H.: Visualizing privacy implications of access control policies in social network systems. Technical Report 2009-927-06, University of Calgary (May 2009)