A Privacy Preservation Model
for Facebook-like Social Network Systems

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

Technical Report CS-2008-05
April 2009

Department of Computer Science
University of Regina
Regina, Saskatchewan, S4S 0A2
Canada

ISBN 978-0-7731-0659 (print)
ISBN 978-0-7731-0660 (online)

A Privacy Preservation Model for Facebook-Like Social Network Systems

Philip W. L. Fong and Mohd Anwar
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
{pwlfong,manwar}@ucalgary.ca

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

April 2009

Abstract

Recent years have seen unprecedented growth in the popularity of social network systems, with Facebook being an archetypical example. Due to the distributed nature of access control in Facebook-style social network systems, it is difficult for a user to anticipate the privacy consequence of such actions as modifying a privacy setting or befriending another user. This work takes a first step in addressing this challenge, 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, on the one hand, are not currently supported by Facebook, but on the other hand possess rich and natural social significance. This work thus deepens our understanding of the design space of privacy preservation mechanisms for 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, 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 [5], a "social network site" is characterized by three functions (our paraphrase): (1) these web services allow users to construct public or semi-public representation of themselves, usually known as user profiles, in a structured 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 social network systems.

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 protect her privacy. Many existing social network systems 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, the richness of the access control mechanism 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 [9, 14]. This makes it difficult for users to fully comprehend the privacy consequence of adjusting their privacy settings or befriending another user. A three-pronged research agenda is thus needed to alleviate this problem: (a) a deeper understanding of the access control paradigm adopted by Facebook, such as delineating the design space of access control mechanisms induced by this paradigm, (b) an articulation of the security requirements for social network systems, by, for example, formalizing the security properties that should be enforced by systems sharing the same access control paradigm as Facebook, and (c) effective means to help users assess the privacy consequence of her actions, an endeavor that traditionally belongs to the domain of safety analysis [11, 15, 22, 20], or, more recently, security analysis [13, 14].

This work is a preliminary step to address 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 social network systems. 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 social network systems, 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 essentially a form of history-based access control [23], with two specializations. Firstly, the form of history tracked by Facebook is the ***communication history*** between pairs of users. Authorization decision is a function of such a history. Secondly, to make such an authorization model computationally tractable and socially rooted, the global communication history is ***abstracted***, in the sense of Fong [8], into a social graph, the topology of which becomes the basis of authorization decisions.

2. We formalized the above insight into a concrete access control model for delimiting the design space of access control mechanisms in Facebook-style social network systems. 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 (to be explained in Sect. 4). We argue 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 and history-based access control policies that possess rich and natural social significance: e.g., degree of separation, known quantity, clique, trusted referral, and staged acquaintance. We thus argue that the design space induced by our access control model should be considered in future design of social network systems.

This paper is organized as follows. Sect. 2 surveys related literature. Sect. 3 provides a high level analysis of the access control mechanism of Facebook, as well as highlights of its distinctiveness and possible generalization. Sect. 4 defines an access control model that captures the above-mentioned distinctiveness and generalization. In Sect. 5, 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. Related work, future work and conclusions are given in Sect. 6.

2 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 in that issue contains a survey of privacy and security issues in Social Network Systems [5]. An early work on social network privacy is [10], which highlights potential privacy attacks on social networks. [1] studies the relationship between user demographics and privacy settings in social networks. Phishing attacks on social networks are discussed in [12]. The privacy impact of the News Feed feature of Facebook is studied in [4]. The lack of flexibility of existing privacy settings in social networks is analyzed in [19]. There is a growing body of literature on the anonymization of social networks (e.g., [24, 17]).

To the best of our knowledge, this is the first work to provide a formal articulation of the access control model behind the Facebook privacy preservation mechanism. We argue in Sect. 3.2 that the access control paradigm behind Facebook is distinct from capability systems, [6, 16], discretionary access control [9, 14] and role-based access control [21, 7]. We also compared this access control paradigm to history-based access control [23] by identifying the history information consumed by the authorization mechanism. Consequently, our work is related to [8]. While both [8] and this work employs 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.

3 Access Control in Facebook and Beyond

3.1 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" [6, 16] of the user in the system, through which access is mediated[1]. There are two means by which a profile may be reached — global name search and social graph traversal.

[1]Facebook would not be considered by the object capability community to be a pure capability system due to the existence of global name search [16].

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. Not every accessing user sees the same profile items when a profile 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 form of 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 once her search listing is reached.

Policies We have seen in the above discussion that various aspects of user activities are control by user-specified policies (e.g., search policy, access policy, etc). This is typical of a discretionary access control systems [9, 14], in which a user may grant access privileges to other users. Facebook offers a fixed set 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 [16]. 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 the 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.

3.2 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 [6, 16]. 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.

D2 *Relation-Based Policies.* Due to the lack of a global name space for accessible resources (a feature very common to capability systems [16]), 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 Discretionary Access Control [9, 14], nor to a centrally administrated set of roles, as in the case of Role Based Access Control [21, 7]. Instead, privileges are granted with respect to an intentionally-specified relation, the articulation of which is administrated in a distributed manner.

D3 *Abstraction of Communication History into a Social Graph.* Facebook is a typical History-Based Access Control system [23]. Authorization 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 [8], into a social graph, the topology of which becomes the basis of authorization decisions.

Generalization Facebook embodies the aforementioned 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 Social Network Systems, 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 delineate the design space of access control mechanisms embodying such a paradigm.

4 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., $\mathcal{G}(S) = \{\, \langle S, E\rangle \mid E \subseteq \mathcal{P}_2(V) \,\}$). We use the the standard λ-notation for constructing anonymous functions [18]: i.e., $\lambda x\,.\,e$ is the function with formal parameter x and body expression e. We write $S \rightharpoonup T$ for the set of all partial functions

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

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 $(\lambda x \,.\, \mathsf{if}\ x = s\ \mathsf{then}\ t\ \mathsf{else}\ f(x))$. We represent records by tuples with named fields: $\langle \mathit{field}_1, \ldots, \mathit{field}_n \rangle$. If t is such a tuple, we write $t.\mathit{field}_i$ to denote the component corresponding to the field field_i.

4.1 System

Our model defines a family of Facebook-inspired ***Social Network Systems (SNSs)***. Every member of the family represents a point in the design space of access control mechanisms represented by our model.

4.1.1 Basic Ontology

A SNS is made up of ***users*** and ***objects*** (aka profile items). Users are uniquely identified by ***user identifiers***, which 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 identifier $u \in Sub$ 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***.

4.1.2 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 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. Knowing who initiates an event allows us to define what it means to have the consent of a party. This concept will become useful in our future work (Sect. 6).

For the ease of addressing users in the following discussion, we assume that the set of users is totally ordered by $\prec$. For each pair of users $\{u, v\}$, we define an identification function $\iota_{\{u,v\}} : \{u, v\} \to \mathbb{B}$ 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 pair. The inverse $\iota^{-1}_{\{u,v\}}$ translates Boolean identifiers back to the users they represent. The set of communication events between two given users can be uniquely identified by the members of $\mathbb{B} \times \Sigma$, such that the ordered pair $(\iota_{\{u,v\}}(u), a)$, where $a \in \Sigma$, uniquely identifies the initiator to be u 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 [23] to model the communication protocol between user pairs, as well as to track communication history. We reuse the notational convention in [8]. A ***communication automaton (CA)*** is a quadruple $M = \langle \Sigma, Q, q_0, \delta \rangle$, where

- Σ is a countable set of communication primitives.
- Q is a countable set of ***communication states***.
- $q_0 \in Q$ is a distinguished ***start state***.
- $\delta : Q \times \mathbb{B} \times \Sigma \rightharpoonup Q$ 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 $His : \mathcal{P}_2(Sub) \to Q$, 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.

4.1.3 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***, $Adj : Q \to \mathbb{B}$, 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:

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

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.

4.1.4 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 following signature:

$$Sub \times Sub \times \mathcal{G}(Sub) \times Q \to \mathbb{B}$$

Given an object owner, an object accessor, the current social graph, as well as the current communication state 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.

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:

$$\lambda(u, v, G, q)\,.\,P_1(u, v, G, q) \vee P_2(u, v, G, q)$$

The policy predicates $P_1 \wedge P_2$ and $\neg P_1$ can be defined similarly. We also define $\top$ and $\bot$ to be the policy predicates that always return true and false respectively.

4.1.5 User-Specified Policies

A SNS allows users to specify four types of access control policies:

1. Every user u may specify a ***search policy*** (i.e., a predicate of the type $Sub \times Sub \times \mathcal{G}(Sub) \times Q \to \mathbb{B}$), 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 reaches 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 the neighbors of u 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 should be allowed to initiate communication primitive a with u as the receiver once v manages to reach the search listing of u.

4. Every user u may assign an ***access policy*** to each of the objects it owns. Such a policy specifies if an accessor v may access that object once v manages to reach the search listing of user u.

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

4.1.6 System

A ***social network system (SNS)*** is an pentuple $N = \langle Sub, Obj, M, Adj, \mathcal{PS}\rangle$:

- Sub is a finite set of ***user identifiers***.
- Obj is a finite set of ***object identifiers***. It is assumed that each object in the system is uniquely identified by an ordered pair in $Sub \times Obj$.
- $M = \langle \Sigma, Q, q_0, \delta\rangle$ is a CA.
- $Adj : Q \to \mathbb{B}$ is an ***adjacency predicate***.

- $\mathcal{PS} = \{\mathcal{PS}_r\}_{r \in Res}$ is a family of ***policy spaces*** indexed by ***resources*** $r \in Res$, such that $Res = \{\mathsf{search}, \mathsf{traversal}\} \cup \Sigma \cup Obj$, and each $\mathcal{PS}_r$ is a countable set of policy predicates (i.e., with type signature $Sub \times Sub \times \mathcal{G}(Sub) \times Q \to \mathbb{B}$). Intuitively, $\mathcal{PS}_{\mathsf{search}}$ specifies the set of policy predicates that users may legitimately adopt as their search policies, while $\mathcal{PS}_{\mathsf{traversal}}$, $\mathcal{PS}_a$ and $\mathcal{PS}_o$ specify, respectively, the set of legitimate traversal policies, the set of legitimate communication policies for communication primitive a, and the set of legitimate access policies for object type o.

Notice that users in a system is not free to choose any policy they want. They must select policies built into the system. The design of the policy spaces is thus a fundamental component of the protection system.

4.2 System States

4.2.1 State

Suppose a system $N = \langle Sub, Obj, M, Adj, \mathcal{PS} \rangle$ is given such that $M = \langle \Sigma, Q, q_0, \delta \rangle$. A ***state*** of N is a pair $S = \langle His, Pol \rangle$:

- $His : \mathcal{P}_2(Sub) \to Q$ maps each pair of users to their current communication state. We also write $His^\sharp : \mathcal{P}_2(Sub) \cup \mathcal{P}_1(Sub) \to Q$ to denote the function $(\lambda\{u, v\}\,.\,\mathsf{if}\ u = v\ \mathsf{then}\ q_0\ \mathsf{else}\ His(\{u, v\}))$. That is, $His^\sharp$ is the extension of His that maps $\{u, v\}$ to q_0 whenever $u = v$.
- $Pol : Sub \times Res \to \bigcup_{r \in Res} \mathcal{PS}_r$, such that $\forall u \in Sub\,.\,\forall r \in Res\,.\,Pol(u, r) \in \mathcal{PS}_r$, is a mapping that records the current policy for every resource of every user.

4.2.2 Reachability

Fig. 1 describes the rules for navigating the social graph. Specifically, the following sequent holds whenever accessor v is permitted to navigate the social graph to reach the search listing of user u.

$$S \vdash_N v\ \mathsf{finds}\ u$$

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 visibility of ones search listing is an important component of protection.

4.2.3 Access

Fig. 2 specifies the rules for object access. Specifically, the following sequent holds whenever accessor v is permitted to access object o of owner u:

$$S \vdash_N v\ \mathsf{reads}\ u.o$$

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).

$$S \vdash_N u \text{ finds } u \qquad \text{(F-Slf)}$$

$$\frac{G = \mathsf{SG}(N.Adj, S.His) \quad \{u, v\} \in E(G)}{S \vdash_N v \text{ finds } u} \qquad \text{(F-Frd)}$$

$$\frac{S \vdash_N v \text{ finds } u' \quad G = \mathsf{SG}(N.Adj, S.His) \quad q = S.His^\sharp(\{u', v\}) \quad \{u, u'\} \in E(G) \quad S.Pol(u', \mathsf{traversal})(u', v, G, q)}{S \vdash_N v \text{ finds } u} \qquad \text{(F-Trv)}$$

$$\frac{G = \mathsf{SG}(N.Adj, S.His) \quad q = S.His^\sharp(\{u, v\}) \quad S.Pol(u, \mathsf{search})(u, v, G, q)}{S \vdash_N v \text{ finds } u} \qquad \text{(F-Sch)}$$

Figure 1: Definition of $S \vdash_N v$ finds u.

$$\frac{S \vdash_N v \text{ finds } u \quad G = \mathsf{SG}(N.Adj, S.His) \quad q = S.His^\sharp(\{u, v\}) \quad S.Pol(u, o)(u, v, G, q)}{S \vdash_N v \text{ reads } u.o} \qquad \text{(R-Acc)}$$

Figure 2: Definition of $S \vdash_N v$ reads $u.o$.

4.3 State Transition

The protection state of a system is changed by a fixed set of transition rules. To allow us to refer to these transitions, we define a set Δ of transition identifiers, the syntax[3] of which is given in Fig. 3. 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. 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.

Following standard practice, we write $S \xrightarrow{w}_N S'$ for $w \in \Delta^*$ whenever S can transition to S' through the sequence of transitions identified by w.

[3]The syntax of transition identifiers is not context free. We use a CFG-like notation to describe its syntax anyway.

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

Figure 3: Transition identifiers.

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

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

Figure 4: Definition of $S \xrightarrow{t}_N S'$.

4.4 Monotonicity, Propriety and Definability

A policy predicate P is said to be ***monotonic*** iff

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

for every $u,v \in Sub$, $G \in \mathcal{G}(Sub)$, $e \in \mathcal{P}_2(Sub)$, and $q \in Q$. 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. Note that monotonicity is preserved by the policy combinators $\wedge$ and $\vee$, but not necessarily by $\neg$. Conversely, a policy predicate P is said to be ***anti-monotonic*** iff

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

for every $u,v \in Sub$, $G \in \mathcal{G}(Sub)$, $e \in \mathcal{P}_2(Sub)$, and $q \in Q$. As expected, $\neg P$ is monotonic if P is anti-monotonic. A system is said to be monotonic iff each of its policy spaces is made up of monotonic policies only. Monotonicity is consistent with our topological intuition: as the social graph becomes denser, access becomes easier. From now on we consider only monotonic systems.

A system state S_0 is said to be a ***proper initial state*** whenever the following conditions are met:

1. The communication state between every pair of users is q_0.
2. Every user-specified policy comes from the appropriate policy spaces.
3. The sequent $S_0 \vdash_N v \ \mathsf{finds}\ u$ is false whenever $u \neq v$. (Consequently, $S_0 \vdash_N v \ \mathsf{reads}\ u.o$ is false whenever $u \neq v$.)

For the above to be feasible, the system must satisfy the following conditions:

- $Adj(q_0) = 0$. (Consequently, F-Frd is rendered inapplicable.)
- $\mathcal{PS}_{\mathsf{search}}$ contains at least one policy predicate that returns 0 when the social graph has no edge or when the communication state is q_0. (Consequently, F-Sch can be rendered inapplicable.)

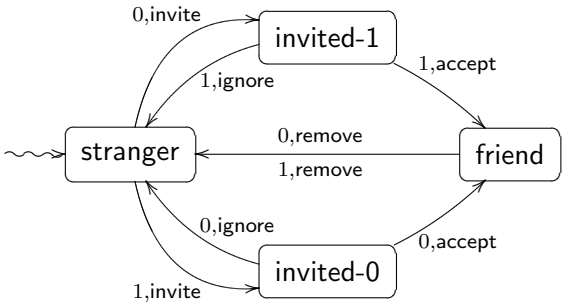

Figure 5: Transition diagram for the communication automaton of $\mathcal{FB}_{lite}$.

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

A state S is said to be ***definable*** iff it is reachable from some proper initial state S_0. That is, $S_0 \xrightarrow{w}_N S$ for some $w \in \Delta^*$. We consider only definable states in the sequel.

5 Sample Instantiations

In this section, we illustrate the utility of our model by demonstrating how it can be instantiated to concrete SNSs.

5.1 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. More importantly, our goal here is to illustrate the point 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 $\mathcal{FB}_{lite} = \langle Sub, Obj, M, Adj, \mathcal{PS} \rangle$, where

- Sub is the set of all user identifiers.
- Obj is the following set:

  $\{$Basic-Information, Contact-Information, Personal-Information, Status-Updates, Wall-Posts, Education-Info, Work-Info$\}$

- $M = \langle \Sigma, Q, q_0, \delta \rangle$, where
  - $\Sigma = \{$invite, accept, ignore, remove$\}$
  - $Q = \{$stranger, invited-1, invited-0, friend$\}$
  - $q_0 =$ stranger
  - δ is defined as in Fig. 5.
- $Adj = (\lambda q \,.\, q =$ friend$)$

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

Figure 6: A list of Facebook-inspired policy predicates.

a	$\mathcal{PS}_a$
accept	{everyone}
ignore	{everyone}
remove	{everyone}
invite	{no-one, friends-of-friends, everyone}

Figure 7: Communication policy spaces for $\mathcal{FB}_{lite}$.

- $\mathcal{PS}_{\mathsf{traversal}} = \{\mathsf{no\text{-}one}, \mathsf{only\text{-}me}, \mathsf{only\text{-}friends}, \mathsf{friends\text{-}of\text{-}friends}, \mathsf{everyone}\}$, where the policy predicates are defined in Fig. 6.

- $\mathcal{PS}_{\mathsf{search}}$ could have been defined in the same way as $\mathcal{PS}_{\mathsf{traversal}}$ 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:

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

  This predicate returns true iff u has extended a friendship invitation to v. Then $\mathcal{PS}_{\mathsf{search}}$ is defined as follows:

  $$\{P \vee \mathsf{owner\text{-}invited} \mid P \in \mathcal{PS}_{\mathsf{traversal}}\}$$

  As a result, initiating a friendship invitation will cause the search listing of the initiator to become accessible to the invited party. This usage of $\mathcal{PS}_{\mathsf{search}}$ demonstrates how accessibility exceptions can be handled.

- For a typical $o \in Obj$, $\mathcal{PS}_o$ can be defined to be the same as $\mathcal{PS}_{\mathsf{traversal}}$. 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 $\mathcal{PS}_{\mathsf{Basic\text{-}Information}} = \mathcal{PS}_{\mathsf{search}}$.

- $\mathcal{PS}_a$ is defined in Fig. 7. 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 $\mathcal{PS}_a = \{\mathsf{everyone}\}$ for most a. Secondly, a user may not always want to allow friendship invitations from strangers. $\mathcal{PS}_{\mathsf{invite}}$ is therefore set to $\{\mathsf{no\text{-}one}, \mathsf{friends\text{-}of\text{-}friends}, \mathsf{everyone}\}$

Notice that all the policy predicates involved in the definition above are monotonic.

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

- Groups, networks, 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.

Nevertheless $\mathcal{FB}_{lite}$ illustrates how the model can be instantiated. We believe that reasonable efforts would 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 group members and the group user. A policy similar to friends-of-friends would allow peer group members to access objects owned by one another.

5.2 Topology-based Policies

In the following we explore examples of possible policies other than those already offered by Facebook. The goal is to illustrate the possibilities supported by the proposed model.

We begin by presenting example policies that are based purely on topological information provided by the social graph. 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). Our focus here is on access policies:

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

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

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 q 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 $\mathsf{common\text{-}friends}_k$ be the following predicate:

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

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 $\mathsf{common\text{-}friends}_1 = \mathsf{distance}_2$.

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

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

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-cligue. 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 grant. 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 $\mathsf{clique}_2 = \mathsf{distance}_1$.

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

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

The policy grants access whenever v is a mutual friend of at least k users belonging to a specific user set U. Essentially, some users (U) are considered more trusted intermediaries than others. Acquaintance with them is a license to access. Such a policy can be used in two ways. First, the policy can be employed to assert that some friends are more trusted than others in mediating access. Second, in the case when an interest group is represented as a user, and group membership is represented by being adjacent to the group user, then the above policy can be employed to grant access to those individuals sharing at least k interests. Lastly, note that $\mathsf{common\text{-}friends}_{k,Sub} = \mathsf{common\text{-}friends}_k$.

Stranger Consider $\neg\mathsf{distance}_k$, the negation of $\mathsf{distance}_k$. 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, $\neg\mathsf{distance}_k$ is not monotonic.

5.3 History-based Policies

We continue our focus on access policies, but this time we introduce history-based information to guide authorization decisions.

Given $Q' \subseteq Q$ and $P : Sub \times Sub \times \mathcal{G}(Sub) \times Q \to \mathbb{B}$, let $Q' \rightarrowtail P$ denote the following predicate:

$$\lambda(u, v, G, q) \,.\, q \in Q' \wedge P(u, v, G, q)$$

This policy imposes P when the communication state q between u and v belongs to Q', and deny access otherwise. Such a policy can be combined with the policy combinator $\vee$ to form composite policies conditional on the communication history between two users:

$$(Q_1 \rightarrowtail P_1) \vee (Q_2 \rightarrowtail P_2) \vee \ldots \vee (Q_k \rightarrowtail P_k)$$

where $Q_1, \ldots, Q_k \subseteq Q$, and $P_1, \ldots, P_k$ are policy predicates. The above policy says, if the communication state is q, then apply policy P_1 if $q_1 \in Q_1$, apply policy P_2 if $q_2 \in Q_2$, ..., and reject otherwise.

Staged Acquaintance One can design a consent protocol, in which acquaintance is divided into multiple stages, advancement in stages is based on consent, and progression in stages allows increasingly liberal policies to be used.

Consider the following extension of $\mathcal{FB}_{lite}$. Suppose the state space of the communication automaton contains the states stranger, acquaintance and friend to represent three levels of acquaintance. Suppose further that the adjacency predicate is the usual $(\lambda q \,.\, q = \mathsf{friend})$. The policy

$$(\mathsf{stranger} \rightarrowtail \mathsf{common\text{-}friends}_3) \vee (\mathsf{acquaintance} \rightarrowtail \mathsf{distance}_3) \vee (\mathsf{friend} \rightarrowtail \top)$$

allows access from strangers when they are known by 3 friends of the owner, from acquaintances when they are within 3 degrees of separation, or from friends unconditionally. Notice the relaxation of constraints when one progresses from stranger to acquaintance and then to friend.

6 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 are interested in addressing challenge (b), the identification of security properties that should be enforced in instantiations of our SNS model, and challenge (c), the design of analytic tools to help users anticipate the privacy consequence of their actions. Another direction is to further generalize the model to account for richer forms of acquaintance relations and policies, including user-defined, asymmetric relations, and ostensionally specified policies[4].

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

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

[4] That is, specification by pointing out examples.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

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

[13] 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.

[14] 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.

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

[16] 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.

[17] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Proceedings of the 2009 IEEE Symposium on Security and Privacy (S&P'09)*, Oakland, California, USA, May 2009. To appear.

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

[19] 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.

[20] Ravi S. Sandhu. The typed access matrix model. In *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, pages 122–136, 1992.

[21] 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.

[22] 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.

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

[24] Brian Thompson and Danfeng Yao. The union-split algorithm and cluster-based anonymization of social networks. In *Proceedings of the 4th International Symposium on ACM Symposium on Information, Computer and Communications Security (ASIACCS'09)*, pages 218–227, Sydney, Australia, March 2009.