Relationship-based Access Control for Online Social Networks: Beyond User-to-User Relationships

Yuan Cheng, Jaehong Park and Ravi Sandhu
Institute for Cyber Security
University of Texas at San Antonio
ycheng@cs.utsa.edu, jae.park@utsa.edu, ravi.sandhu@utsa.edu

***Abstract*—User-to-user (U2U) relationship-based access control has become the most prevalent approach for modeling access control in online social networks (OSNs), where authorization is typically made by tracking the existence of a U2U relationship of particular type and/or depth between the accessing user and the resource owner. However, today's OSN applications allow various user activities that cannot be controlled by using U2U relationships alone. In this paper, we develop a relationship-based access control model for OSNs that incorporates not only U2U relationships but also user-to-resource (U2R) and resource-to-resource (R2R) relationships. Furthermore, while most access control proposals for OSNs only focus on controlling users' normal usage activities, our model also captures controls on users' administrative activities. Authorization policies are defined in terms of patterns of relationship paths on social graph and the hopcount limits of these path. The proposed policy specification language features hopcount skipping of resource-related relationships, allowing more flexibility and expressive power. We also provide simple specifications of conflict resolution policies to resolve possible conflicts among authorization policies.**

I. Introduction

Online social networks (OSNs) have attracted a large amount of users to regularly connect, interact and share information with each other for different purposes. Users share a tremendous amount of content with other users in OSNs using various services. The explosive growth of sensitive or private user data that are readily available in OSNs has raised an urgent expectation for effective access control that can protect these data from unauthorized users in OSNs.

Access control in OSNs is typically based on the relationships among users in the social graph. That is, granting access to an accessing user is subject to the existence of a direct or indirect relationship of certain types between the accessing user and the controlling users of the target. Many existing OSN systems enforce a rudimentary and limited relationship-based access control mechanism, offering users the ability to choose from a pre-defined policy vocabulary, such as "public", "private", "friend" or "friend of friend". Google+ and Facebook recently introduced customized relationships, namely "circle" and "friend list", providing users richer options to differentiate distinctly privileged user groups. Meanwhile, more sophisticated relationship-based access control models have been proposed by researchers (see related works section). These proposals explore more flexible and expressive solutions than provided by current commercial OSNs, such as supporting multiple relationship types in policy languages or taking the trust value of relationships into account in the control of information dissemination. One common characteristic found in most of these commercial and academic solutions is that they mainly focus on user-to-user (U2U) relationships between accessing user and the resource owner, and at least implicitly assume ownership is the only manifestation of user-to-resource (U2R) relationships. However, this is not sufficient to capture many user activities found in today's OSN applications, where users can perform actions that create relationships between users and resources other than ownership. For example, tagging a friend on a photo will create U2R relationship between the photo and the tagged user which consequently may allow friends of the tagged user to access the photo. Hence the tagged user may want to control other related users' access to the photo. Likewise, users' actions can establish resource-to-resource (R2R) relationships such as photos under the same album, comments to a blog post, etc. To enable fully expressive relationship-based access control, it is necessary to exploit U2R and R2R relationships in addition to U2U relationships for authorization policies and decisions.

In OSN, users are allowed to configure access control policies for their own content and activities. Allowing U2R relationship-based access control further enables users to specify policies for contents related to them and activities of other related users. Since a change of relationships may result in a change of authorization, the creation and termination of relationships needs to be treated differently from usage activities to normal resources. Thus, access control in OSNs has to address the management of access control policies and relationships in addition to normal usage activities by means of U2U, U2R and R2R relationships. Although Carminati et al [6], [7] introduced a framework that allows system administrators to specify administrative policies in ontology-based representations, they did not provide a policy management model for managing policies and resolving policy conflicts. Most of the other relationship-based access control models do not incorporate users' administrative activities.

Since multiple users can express access control policies for a user or a resource, it is expected that there will be several policies applicable to the same access request which will inevitably raise conflicts. For example, Bob sets his policy so that he can get friendship request from anyone in the system, while at the same time policies defined by his parents may only allow him to receive such request from his friends of friends. To resolve such conflicts, it is necessary to introduce conflict resolution policies, which are (meta-)policies about how authorization policies are to be interpreted and how policy conflicts are resolved.

In our previous work [11], we developed an access control model for OSNs based on U2U relationships, using regular expression notation in its policy specifications to express path patterns between the accessing user and the controlling user of the target. This model only addresses access control over normal usage activities. We presented a graph traversal algorithm for path checking with correctness proof and complexity analysis. Building on this prior work, in this paper we introduce a relationship-based access control model that utilizes not only U2U relationships but also U2R and R2R relationships in its scope. The proposed model covers users' normal usage activities as well as administrative activities, which have only been rarely addressed in the existing literature. The regular expression based policy specification language of [11] is extended here to be flexible and expressive enough to support various U2U, U2R and R2R relationship-based access control policies.[1] We incorporate simple system-defined conflict resolution policies as a default solution for the inevitable authorization policy conflicts. By including U2U, U2R and R2R relationships in authorization decision process and controls on users' administrative activities, the proposed model significantly extends the previous U2U relationship-based access control model of [11].

The remainder of this paper is organized as follows. In section II, we provide a brief discussion of related research works. Section III discusses the limitation of control based on U2U relationships and taxonomy of user's access types in OSN with U2U, U2R and R2R relationships. Section IV describes the components of the proposed relationship-based access control model in OSNs. In section V, we further present the model that captures usage activities and administrative activities as well as specification language for authorization policies, and describe conflict resolution policies that can be used to resolve conflicts. Section VI discusses some use cases we identified in the model. Section VII outlines some future work and concludes the paper.

II. Related Works

In the following, we review some of the related works that focus on access control mechanisms for OSNs and policy conflict resolution in access control systems.

A. Access Control Models for OSNs

Inspired by research in trust and reputation systems, some early solutions proposed by Kruk et al [21] and Carminati et al [8], [9] identified aggregated trust value, denoting the level of relationship, along with relationship type and depth on a path from the resource owner to the accessing user as parameters for authorization. While Kruk's work only considers one relationship type, Carminati's work allows multiple relationship types but only supports trust computation of a relationship path of a single type at a time. Carminati et al also proposed a semi-decentralized architecture, where access rules are specified in terms of relationship type, depth and trust metrics by individual users in a discretionary way [10]. The system features a centralized certificate authority to assert the validity of relationship paths, while access control enforcement is carried out on the decentralized user side.

A formal model for access control in Facebook-like systems was developed by Fong et al [14], which treats access control as a two-stage process, namely, reaching the search listing of the resource owner and accessing the resource, respectively. Reachability of the search listings is a necessary condition for access. Although lacking support for directed relationships, multiple relationship types and trust metric of relationships, this model allows expression of arbitrary topology-based properties, such as "k common friends" and "k clique", which are beyond what Facebook and other commercial OSNs offer.

Fong et al [15] proposed a formal ReBac [17] model for social computing applications, which employs a modal logic language for policy specification and composition. In [16], Fong et al later extended the policy language and studied its expressiveness. These two models allow multiple relationship types and directional relationships. Authorization are based on U2U relationships between the accessing user and the resource owner, and relationships are articulated in contexts.

In [6], [7], Carminati et al proposed an access control framework which utilizes relationships among users and resources as the basis for access control and employs the Semantic Web Rule Language (SWRL) to define authorization, administration and filtering policies. Our model proposed in this paper offers more complete policy administration by addressing policy management and conflict resolution. Another semantic web-based approach proposed in [22] allows both users and the system to express policies based on access control ontologies.

B. Policy Conflict Resolution

There is substantial literature on conflict resolution of access control policies, especially in distributed systems, database systems and collaborative environments. Simultaneous presence of conflicting policies can be resolved by various strategies, such as permissions-take-precedence [19], [20], denials-take-precedence [5], [19], [20], specificity precedence [4], [13], recency precedence, strong authorization overriding weak authorization [3], [4], [26], or explicit specification of policy priority [2], [12], [27], etc. Most conflicts discussed in this literature are conflicts between positive and negative authorizations (permissions vs. prohibitions) typically arising due to generality or specificity of the applicable policy in a hierarchy. However, in OSNs possible policy conflicts will likely arise due to policies specified by distinct users carry contrasting authorization.

[1]For the enhanced policy specification language developed in this paper, we need a similarly enhanced graph traversal algorithm for path checking in the new model along with proofs of correctness and complexity. Due to space limitations this is outside the scope of this paper.

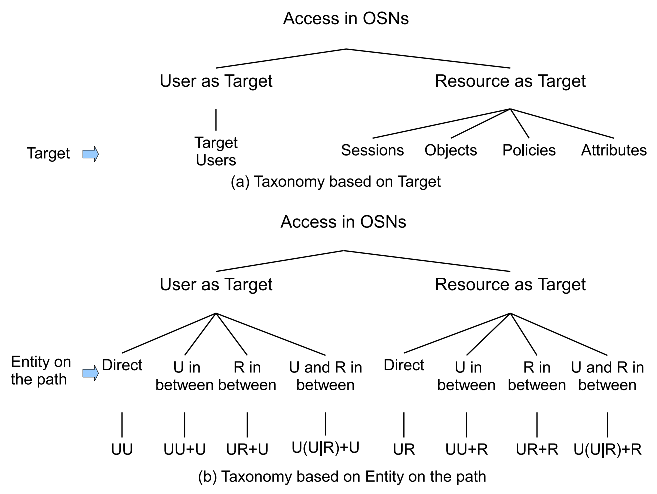

Fig. 1. Access in OSNs

In OSN systems, as long as each user can specify individual policies, policy conflicts become inevitable. [28] applied game theory to a solution for collective policy management in OSNs, where data resources may belong to multiple users. [18] proposed a formal model to address multi-party access control in OSNs with a policy conflict resolution mechanism based on voting scheme to deal with collaborative policies. In this approach, the release of a resource depends on the sensitivity scores assigned by each controlling user and the chosen decision making strategy, such as setting a sensitivity threshold, owner-overrides and full-consensus-permit. Although policy conflict resolution is not the main focus of this paper, it is necessary to explicitly express an unambiguous strategy, whether a conjunction, a disjunction or a prioritized order of relationships between the policy specifiers and the user or resource the policies apply to.

III. Beyond U2U Relationship-based Access Control

In this section, we discuss limitations of U2U relationship-based access control and build a taxonomy of user's access types in OSNs based on U2U, U2R and R2R relationships.

A. Limitation of U2U Relationship-based Access Control

In OSNs, users are encouraged to create profiles, add content onto their pages (e.g., photos, videos, blogs, status updates and tweets), and share these resource objects with other peers. OSNs offer their users various types of user interaction services, including chatting, private messaging, poking and social games. As OSN systems mature, various types of resources need to be protected, such as user sessions, relationships among users and resources, access control policies and events of users. As shown in Figure 1(a), users can launch access requests against both resources (e.g., view a photo or create an access control policy) and users (e.g., invite another user to a game or poke another user).

Social graph represents a global mapping of all individual users and how they are connected in an OSN, where user is a node and a relationship between users is an edge. Access control in most existing OSNs are based on the topology of the social graph, so-called relationship-based access control. Typically, granting access permission to an accessing user is subject to the existence of a particular relationship or a particular sequence of relationships between the accessing user and the target user/resource owner, and access control policies are specified in terms of such U2U relationships. When a user requests access to a resource, current OSNs rely on an implicit relationship, namely ownership, between the resource and its owner, hence the authorization of such U2R access is still based on the underlying U2U relationships.

However, due to the various functionality offered by today's OSNs, there exist several different types of relationships between users and resources in addition to ownership. Consider an example where Bob posts a photo that contains Alice and Carol's images in it and tags them. OSNs usually allow only the owner Bob to have control on who can view the photo, regardless of whether or not Alice and Carol may wish to release their images. To enable Alice and Carol control capability on the photo, their relationships with the photo, which is not ownership, should be considered for authorization purposes. After the photo has been shared by Bob's friends several times, more and more users from different neighborhoods in the network come to view the photo and comment on it. When Dave reads through all the comments in Bob's photo and becomes curious about another user Eve who has commented recently, he decides to poke her to say hello. In this case, Dave and Eve are connected through the photo, not through another user (such as the owner of the photo Bob). Also, users may share or like the blog posts or videos posted by others, and gain the ability to determine how the shared/liked copy of the original content or the fact of sharing and liking activities can be seen by others. Consider another scenario where Betty finds a weblink originally posted by Ed interesting and then shares it with her friends. From her activity, she acquires the ability to decide how the weblink can be available to others. As users get increasingly involved in these activities in OSNs, current U2U relationship-based access control mechanism is not able to offer the appropriate control and requires extensions to bring U2R and R2R relationships into consideration.

In recent years, Facebook has gradually expanded the idea of social graph to so-called Open Graph as it launches new services such as photos and places, and includes these in the graph over time. Recently even further extensions to incorporate arbitrary activities and objects are being pushed so as to codify user behaviors effectively. These recent trends in commercial OSNs strengthen our belief that it is useful to include resources, such as objects and activities, in the social graph. By means of such an extended social graph, users and all of the resources related to users are interconnected through U2U, U2R, and even R2R relationships, allowing stronger expressive power of relationship-based access control policies.

B. Taxonomy of Access Scenarios

As shown in Figure 1(b), in OSN, a user can access other users (user as a target) or resources (resource as a target). By means of U2U, U2R and R2R relationships, an accessing user and a target user can have a direct relationship or indirect relationships with user(s) in between, resource(s) in between or user(s) and resource(s) in between. Likewise, an accessing user and a target resource can also be characterized in terms of the entities on the relating path.

In the first two cases of accessing a target user, there is no resource involved. An accessing user should either have a particular direct U2U relationship (shown as UU) or a particular sequence of U2U relationships (shown as UU+U)[2] with the target user. Examples of such access to a target user are that Alice's direct friends can poke her, and Bob's friends of friends can request friendship invitation to him. If resources are introduced onto the path between the accessing user and the target user, it brings in U2R and R2R relationships and leads to other possible combinations of relationships. For instance, "Users who are tagged in the same photo can visit each other's profile even though they are not friends" is a typical UR+U example that can be found in OSNs, in which case the photo actually links two unconnected users together. We can make even more complicated policies by connecting users through both users and resources (shown as U(U|R)+U). In the previous example, friends of one of the tagged users may be able to access another tagged user through their mutual friend and the photo.

Similarly, a user may access a resource that directly relates to her (shown as UR), or may find a resource through one or more users in the network (shown as UU+R). Most of the current commercial OSNs and the prior work [6], [10], [14]–[16] deal with these two cases with an implicit assumption of the existence of "own" relationship. When a user requests an access against resource, the system checks if there is a qualified relationship between the accessing user and the resource owner, and then determines the authorization. We believe that it is useful to distinguish different types of U2R relationships, such as "tag", "share" and "like", in the policy specifications rather than relying only on "own". Incorporating R2R relationships and connecting resources on the path enables UR+R and U(U|R)+R cases, so that users may be able to access some resources that are connected to users' related resources. For example, if user Alice is tagged in one of Bob's photo, then Alice may get the privilege to view other photos in the same album without being Bob's contact of any type.

The above discussion indicates that allowing U2R and R2R relationships in access control gives users more complete and flexible expressive power than the currently prevailing U2U-only approaches.

IV. Relationship-Based Access Control Model Components and Characteristics

In this section, we identify the components of the proposed relationship-based access control model for OSNs, and discuss crucial characteristics of the model.

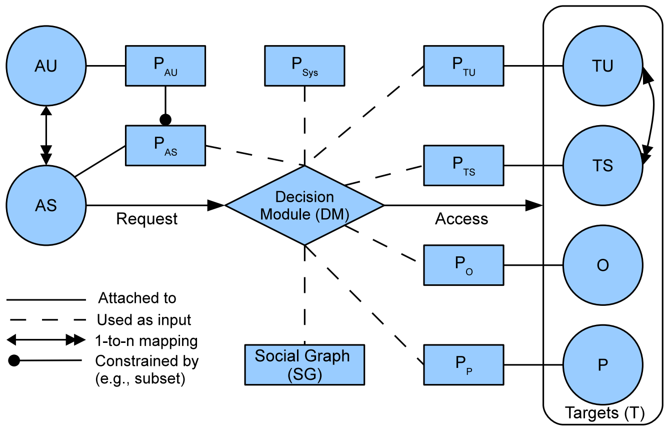

Fig. 2. Model Components

A. Components

Figure 2 shows a conceptual diagram for the relationship-based access control model. It comprises six categories of basic components: users, sessions, resources, policies, social graph and decision module, as discussed below.

Users. A user represents a human being registered in an OSN system to whom authorization may be granted. Users maintain relationships with each other, own a number of resources, and perform various kinds of actions against resources and users in the system. Users can be identified as accessing users (AU) and target users (TU), based on the roles they play with respect to access. Accessing users are users who perform certain types of access against targets, carrying authorization policies (Accessing User Policies or P_{AU}). Target users are users against whom access is performed. Target users also carry authorization policies (Target User Policies or P_{TU}).

Sessions. A session is an active instance of a user who has logged into the OSN. Accessing users perform access through sessions (AS), while target users may or may not have a session instance (TS) at the time of access. In other words, some accesses can be placed only when a target user is online (e.g., chatting) while other accesses do not require this and can be placed on a target user who is not logged in at the time of the accesses (e.g., poking). The user-session distinction allows sessions to have different policies and attributes from those of the corresponding user by partially inheriting them possibly along with some other policies and attributes. This is depicted in Figure 2 as "constrained by". A user can have multiple sessions with differing access control policies, while a session is only associated with a single user. In general, all sessions are considered objects and can be created, suspended, or resumed by another session or by the user. Sessions are also called subjects in the access control literature. The term session was introduced in the role-based access control literature and has become widely used in that context.

Resources. Resources are non-user targets to be protected in access. They include target user's sessions (TS), objects (O) users shared in the system as well as access control policies (P). Because resources are not human beings, access control

[2]Note, "+" denotes one or more occurrences of the applicable entity (U, R or U|R) on the path.

policies for resources are defined by users who possess the corresponding administrative privileges.

Policies. Access control policies are a set of rules that govern the ability of sessions (subjects) to access targets. Like in many computer systems, OSNs allow the system security administrators to define a central policy that is guaranteed to be enforced for all users and resources in the system, called *system-specified policy* (P_{Sys}). Additionally, users have the ability to express own preferences with respect to themselves or their related users and resources. *Accessing user policy* (P_{AU}), *target user policy* (P_{TU}), *accessing session policy* (P_{AS}), *object policy* (P_O), *policy for policy* (P_P) are defined by users and applied to accessing users, target users, accessing sessions, objects and policies, respectively. System-specified policies consist of two types of policies, namely *authorization policies* and *conflict resolution policies*. Authorization policies allow the system and users to specify who is authorized to exercise which action on the user or resource, while conflict resolution policies specify how conflicts among authorization policies from multiple parties are to be solved. For the purpose of this paper we assume that conflict resolution policies are specified entirely by the system administrators as part of P_{Sys}.

Social Graph (SG). Social graph denote connections among users and resources in the system. Although "relationships" on the graph usually refer to relationships among individual users in many OSN systems in practice and theory, they also can include U2R relationships and R2R relationships as we have argued above. U2U relationships are typically represented by the social graph. We extend the social graph to incorporate U2R relationships and R2R relationships as well, thus forming a network of users, sessions, objects as well as their access control policies.

Decision Module (DM). The access decision module in Figure 2 consolidates all the necessary policies from P_{AS}, P_{TU}, P_{TS}, P_O, P_P and P_{Sys} as well as the relationships on the social graph, and makes a decision at the time of request. Access decision module can handle potential policy conflicts by consulting conflict resolution policies in P_{Sys}.

B. Characteristics

We identify three essential characteristics that need to be addressed by OSN access control models, as follows.

Policy Individualization. As identified in [24], [25], unlike in traditional access control systems, OSNs allow individual users to express their own preferences over access to the content rather than having a single system-wide access control policy defined by the system security administrator. Moreover, users other than the resource owner are also able to configure policies for user and resource related to them. For example, parents of a child want to prescribe a boundary within which their child might perform access, and Alice wants to block her colleagues from seeing the party pictures which contain her image. The system needs to collect all of the related individual policies along with the system-specified policies for making access control decisions.

Policy Administration. In OSN, policy administration becomes very important since allowing individual users to specify policies requires the OSN to ensure that only the right users are authorized to specify policies. Our model enables users to specify policies for other users and resources as long as they meet the relationship requirement stated in the policies for the target policy. For example, the system-specified policy may allow users who have "own" or "tag" relationships with the resource to set the policy for that resource. Then the owner or tagged user can control the resource's policy and later modify it to enable or disable who can access the resource.

User-session Distinction. We know that a session is a process in execution on behalf of a user. A user can have multiple sessions with different sets of privileges by creating different degrees of access control policies with the original user's. The user-session distinction facilitates better security and privacy control by minimizing a session's privilege to an adequate level. It becomes especially useful in OSN environments as more and more smart devices and location-based applications are introduced into OSN world. Users logged in from different devices may have distinct access control policies and thus distinct privileges. Users with location-based services enabled may be offered extra functionality than ordinary users are. Much of the current literature in OSN access control does not distinguish a session from a user. We believe differentiating user and session is crucial for effective access control in OSNs.

V. Relationship-based Access Control Model

In the following, we formally define an access control model for OSNs, expressing authorization policies and conflict resolution policies in terms of relationships existing among users and resources in the system.

A. Model Definition

We begin by identifying each component of the model. U is the current set of users, including accessing users (AU) and target users (TU). Associated with each user is a collection of sessions. S is the current set of sessions, which is composed of accessing sessions (AS) and target sessions (TS). R is the set of resources, including target sessions (TS), objects (O) and access control policies (P). We refer to target users and resources as targets, which are the targets of access.

We write $ACT = \{act_1, act_2, \ldots, act_n\}$ which is the set of OSN supported actions, denoting the access modes a user or a session can execute in the system. Each action is defined in active form with accessing user or session as the actor and target users and/or resources as targets. For each action act_i the passive form act_i^{-1} represents the action from the target's perspective.

The overall set of policies P in the OSN is categorized as follows.

- $P_{AU} \subseteq P$, $P_{TU} \subseteq P$, $P_{AS} \subseteq P$, $P_{TS} \subseteq P$, $P_O \subseteq P$, $P_P \subseteq P$ and $P_{Sys} \subseteq P$ are authorization policies for accessing user, target user, accessing session, target session, objects, policies and system-specified policies, respectively.

- $AP_{Sys} \subseteq P_{Sys}$ and $CRP_{Sys} \subseteq P_{Sys}$ represent system-specified authorization policies and conflict resolution policies, respectively.

The first two user-specified policies P_{AU} and P_{TU} are associated with and specified by the corresponding users, while rest of the user-specified policies are specified by users but associated with the resource or another user. The user who actually specifies a policy for other user or resource is called the controlling user (CU), who has a certain type of relationship with the user/resource to whom the policy applies.[3] Accessing session policies are partially inherited from its corresponding accessing user's policies, but may have additional content specific to the session, such as location. System-specified policies, on the other hand, are expressed by the system and applied to all relevant activities across the system. While a resource is always the target of an access, a user, on the other hand, can participate both as an accessing user or a target user in an access with different access control requirements. Hence, the distinction between the active and passive forms of an action becomes significant. We write act and act^{-1} as the active and passive form of an action act, respectively. Accessing user policies, accessing session policies and system-specified policies are indexed by act, and target user policies, target session policies, object policies and policies for policy are indexed by act^{-1}. Authorization policies specified by multiple users may possibly give conflicting results for a requested access. We introduce system-defined conflict resolution policies (CRP_{Sys}) to make unambiguous decisions for authorization policies specified by multiple parties with conflicting interests. The specification of policies is discussed below in part B of this section.

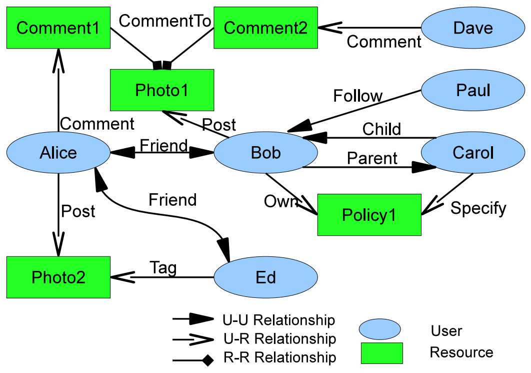

Fig. 3. A Sample Social Graph

As depicted in Figure 3, we abstract an OSN as a directed labeled simple graph, where each vertex represents a user or a resource, whereas each edge corresponds to a relationship among users and resources. The social graph of an OSN SG is formally denoted as a triple $< V, E, \Sigma >$:

[3]In some cases, accessing a user/resource is subject to the relationships existing between the accessing user and the controlling user [11]. See Example 3 in Section VI.

- $V = U \cup R$ is a finite set of vertices on the graph, representing registered users and their resources in the system.
- $\Sigma = \Sigma_{u\_u} \cup \Sigma_{u\_r} \cup \Sigma_{r\_r} = \{\sigma_1, \sigma_2, \ldots, \sigma_n, \sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_n^{-1}\}$, denotes the set of relationship type specifiers in the system. Each relationship type specifier σ is represented by a string of characters. Given a relationship type $\sigma_i \in \Sigma$, we write the inverse of the relationship $\sigma_i^{-1} \in \Sigma$. Relationships are further divided into three categories: U2U (U_U), U2R (U_R) and R2R (R_R) relationships. We express this formally as follows.
  - U_U relationships: $U \times \Sigma_{u\_u} \times U$,
  - U_R relationships: $U \times \Sigma_{u\_r} \times R$ or $R \times \Sigma_{u\_r} \times U$,
  - R_R relationships: $R \times \Sigma_{r\_r} \times R$,

  where $\Sigma_{u\_u} \subseteq \Sigma$, $\Sigma_{u\_r} \subseteq \Sigma$ and $\Sigma_{r\_r} \subseteq \Sigma$.
- $E = E_{u\_u} \cup E_{u\_r} \cup E_{r\_r}$, where $E_{u\_u} \subseteq U \times \Sigma_{u\_u} \times U$, $E_{u\_r} \subseteq (U \times \Sigma_{u\_r} \times R) \cup (R \times \Sigma_{u\_r} \times U)$, $E_{r\_r} \subseteq R \times \Sigma_{r\_r} \times R$, represents the existing relationships among users and resources in the system.

Note that E is defined as directed, since not all of the relationships in OSNs are mutual. For every $\sigma_i \in \Sigma$, there is $\sigma_i^{-1} \in \Sigma$ representing the inverse of relationship type σ_i. Although not explicitly shown on the social graph, we assume the original relationship and its inverse twin always exist simultaneously. Given a vertex $v_1 \in V$, a user $v_2 \in V$ and a relationship type $\sigma \in \Sigma$, a relationship (v_1, v_2, σ) says that there exists a relationship of type σ originating from vertex v_1 and terminating at v_2. There always exists an equivalent form (v_2, v_1, σ^{-1}) at the same time.

It remains to formally define the concept of access request.

- (s, act, T) represents an access request, where $s \in S$ indicates the accessing session, $act \in ACT$ denotes the requested action and $T \subseteq (2^{TU \cup R} - \emptyset)$ gives a non-empty set of target users and resources;

The cardinality and types of the targets are determined by the action.

B. Policy Specifications

The notations used in the policy specification language are defined in Table I, familiar from typical regular expression notation with the addition of hopcount limits and skipping. As described earlier, there are several types of access control policies: accessing user policy, accessing session policy, target user policy, target session policy, object policy, policy for policy, and system-specified policy. Here, system-specified policies comprise authorization policies and conflict resolution policies. System-specified authorization policy allows the system to express access control requirements that apply to the entire set of users or resources, while system-specified conflict resolution policy deals with the potential conflicts of interest among the user-specified authorization policies.

Authorization Policy (AP). Authorization policies are modeled in different formats as shown in Table II. *Accessing User Policy* and *Accessing Session Policy* are represented as a pair $< act, graph\ rule >$ and regulate how an access requester in

TABLE I
POLICY SPECIFICATION NOTATIONS

Concatenation (·)	Joins multiple characters $\sigma \in \Sigma$ or Σ itself end-to-end, denoting a series of occurrences of relationship types.
Asterisk (*)	Represents the union of the concatenation of σ with itself zero or more times. For example, $friend*$ means direct or indirect friends of a user or user herself. $\Sigma*$ is $\bigcup_{i=0}^{\infty} \Sigma^i$, denoting the node itself or nodes with any connection on social graph.
Plus (+)	Denotes concatenating σ one or more times. Similarly for $\Sigma+$.
Question Mark (?)	Represents occurrences of σ zero or one time. $coworker \cdot friend?$ means only coworker or coworker's direct friends can access. Similarly for $\Sigma?$.
Square Bracket ([])	Contains a path rule: a sequence of relationship specifiers with an indicated hopcount limit.
Double Square Bracket ([[]])	Denotes skipping of the path rule contained. The meaning of the skipping feature is discussed in the text.
Disjunctive Connective ($\vee$)	Indicates the disjunction of multiple path specs.
Conjunctive Connective ($\wedge$)	Denotes the conjunction of multiple path specs.
Negation ($\neg$)	Implies the absence of the specified pair of relationship type sequence and hopcount.

TABLE II
AUTHORIZATION POLICY REPRESENTATIONS

Accessing User Policy	$< act, graphrule >$
Accessing Session Policy	$< act, graphrule >$
Target User Policy	$< act^{-1}, graphrule >$
Target Session Policy	$< act^{-1}, graphrule >$
Object Policy	$< act^{-1}, graphrule >$
Policy for Policy	$< act^{-1}, graphrule >$
System Policy for User	$< act, graphrule >$
System Policy for Resource	$< act, o.type, graphrule >$ where $o.type$ is optional

access can behave. Here, act indicates the requested action while $graph\ rule$ denotes the access rule based on social graph. *Target User Policy*, *Target Session Policy*, *Object Policy* and *Policy for Policy* are about how others can perform access on the target, so they use passive form act^{-1} instead of act because the target is always the entity to be accessed, whereas $graph\ rule$ has the same meaning as in the previous policies. *System-specified policies* do not differentiate the active and passive form of an action, since they are not attached to a particular entity in action. However, it is likely that we need to refine the scope of the objects to which the policies apply, thus we bring object types $o.type$ into policy specifications. Hence, system-specified policies are defined in two formats: $< act, graph\ rule >$ for user and $< act, o.type, graph\ rule >$ for resource. Note that $o.type$ is optional and used only if target is an object.

Table III defines the grammar for the graph rules, based on which each graph rule specifies a $startingnode$ and a $pathrule$. Starting node stands for the user or resource where the policy evaluation begins, which can be the accessing user, the controlling user or the target. A path rule is composed of one or more path specs, with each spec stating the required sequence of relationship types and the corresponding hopcount limit for the sequence. Users are allowed to specify a more complicated and fine-grained policy for an action against a target by connecting multiple path specs with conjunctive connective "$\wedge$" and disjunctive connective "$\vee$". Also, negation "$\neg$" over path specs is used to imply the absence of the specified pattern of relationship types and hopcount limit as authorization requirements. Each path spec is denoted as a tuple ($path$, $hopcount$), where $path$ represents from the starting node a sequence of relationship type expressions segmented by "[]" or "[[]]" with local hopcounts, denoting the pattern of relationship types required to grant authorization, whereas $hopcount$ describes the maximum distance between the accessing user and the target on the graph. Within each path segment there is a local $hopcount$ defining the maximum distance requirement for the particular piece of relationship type expression. In some policies, $path$ can be left blank to indicate only the starting node can access. With the use of U2R and R2R relationships, the distance between two users on the graph may be growing significantly. The notion of distance in U2R and R2R relationships is somewhat different from that of U2U relationships. For example, suppose Alice and Bob are friends and Bob owns a photo and Dave is tagged to the photo. Here, the distance between Alice and Bob is 1 and distance of Bob and Dave is 2. While the distance between Alice and Dave is 3, this combined distance is not as meaningful as individual U2U and U2R/R2R distances. Therefore, we may want to omit the distance created by resources by introducing the "skipping" notation, denoted "[[]]", which means that the local hopcount stated inside "[[]]" will not be counted in the global hopcount. For instance, in the path rule "([$f*$,3][[$c*$, 2]], 3)", the local hopcount 2 for $c*$ does not apply to the global hopcount 3, thus allowing $f*$ to have up to 3 hops.

Conflict Resolution Policies (CRP). Due to the nature of policy individualization, multiple policies applicable to authorization of an access request may result in decision conflicts in many scenarios. We assume that policies specified by the system will always be unambiguous so there are no conflicts within P_{Sys}. Conflict resolution policies are then responsible for interpreting how the potential policy conflicts within each category of P_{AS}, P_{TU}, P_{TS} P_O and P_P can be resolved in terms of the precedence or connectives over relationship types. Relationship precedence is used to produce a collective result from multiple policies specified by users with different relationships to the policy holder. To resolve conflicts, we consider three simple and intuitive approaches: disjunctive, conjunctive or prioritized. In a disjunctive approach, satisfying any of the involved policies guarantees access. While for some sensitive contents, it is more meaningful to conjunctively query all the involved policies so that authorization is only allowed by satisfying the requirements of every policy. Whereas, if parental control is facilitated, parents' policies always get priority over children's policies. We write $\vee$, $\wedge$ and $>$ to denote disjunction, conjunction and prioritized order between relationship types, whereas the symbol @ represents a special

TABLE III
GRAMMAR FOR GRAPH RULES

$GraphRule \rightarrow$ "(" $StartingNode$ "," $PathRule$ ")"
$PathRule \rightarrow PathSpecExp \mid PathSpecExp\ Connective\ PathRule$
$Connective \rightarrow \vee \mid \wedge$
$PathSpecExp \rightarrow PathSpec \mid$ "¬" $PathSpec$
$PathSpec \rightarrow$ "(" $Path$ "," $HopCount$ ")" $\mid$ "(" $EmptySet$ "," $HopCount$ ")"
$HopCount \rightarrow Number$
$Path \rightarrow$ ["[" $TypeSeq$ "]" | "[" $TypeSeq$ "," $HopCount$ "]" | "[[" $TypeSeq$ "," $HopCount$ "]]"]+
$EmptySet \rightarrow \emptyset$
$TypeSeq \rightarrow TypeExp$ {"·" $TypeExp$}
$TypeExp \rightarrow TypeSpecifier \mid TypeSpecifier\ Wildcard$
$StartingNode \rightarrow u_a|u_c|t$
$TypeSpecifier \rightarrow \sigma_1|\sigma_2|\ldots|\sigma_n|\sigma_1^{-1}|\sigma_2^{-1}|\ldots|\sigma_n^{-1}|\Sigma$ where $\Sigma=\{\sigma_1,\sigma_2,\ldots,\sigma_n,\sigma_1^{-1},\sigma_2^{-1},\ldots,\sigma_n^{-1}\}$
$Wildcard \rightarrow$ "∗" | "?" | "+"
$Number \rightarrow [0-9]+$

relationship "null" that denotes "self".

Let us consider some examples of conflict resolution policies as follows.

$< read^{-1}, (own \wedge tag) >$

Both the owner's and the tagged users' "$read^{-1}$" policies over the photo are honored.

$< friend\_request, (parent > @) >$

When child attempts friendship request to someone, parents' policies get precedence over child's own will.

$< share^{-1}, (own \vee tag \vee share) >$

A weblink is sharable if either the original owner, or any of the tagged users or shared users allows.

While evaluating an access request, if the decision module discovers two or more opposing policies from the same policy set, it looks up the corresponding conflict resolution policy for the action to determine how to reconcile the conflict. Note that conflict resolution policies only apply to the conflicting policies from the same policy category, the decision module still takes the conjunction of $P_{AS}, P_{TU}, P_{TS}, P_O, P_P$ and AP_{Sys} to make a final decision.

C. Access Evaluation Procedure

Algorithm 1 specifies how the access evaluation procedure works. After a session of a user s requests an act against target(s) T, say (s, act, T), the access decision module first collectively assembles s's session policy about act, a collection of act^{-1} policies from each target in T and the system-wide policies over act and object type, if target is an object. Once all the necessary policies are collected, the decision module extracts each path spec from the graph rules, determines the starting node and the evaluating node, and runs path checking for each path spec using the algorithm similar to one introduced in [11]. The evaluation result of each policy is derived from combining the result of each path spec in the policy. Due to possible conflicts between the results of multiple policies, the decision module looks up the system-defined conflict resolution policies to resolve conflicts and compose the final result, and then determines the access.

Algorithm 1 $AccessEvaluation(s, act, T)$

```
1: (Policy Collecting Phase)
2: $s.P_{AS}(act) \leftarrow s$'s policy for $act$
3: if $(T \cap TU) \neq \emptyset$ then
4:     $T.P_{TU}(act^{-1}) \leftarrow \bigcup_{i=1}^{|T \cap TU|} tu_i.P_{TU}(act^{-1})$
5: if $(T \cap TS) \neq \emptyset$ then
6:     $T.P_{TS}(act^{-1}) \leftarrow \bigcup_{j=1}^{|T \cap TS|} ts_j.P_{TS}(act^{-1})$
7: if $(T \cap O) \neq \emptyset$ then
8:     $T.P_O(act^{-1}) \leftarrow \bigcup_{k=1}^{|T \cap O|} o_k.P_O(act^{-1})$
9: if $(T \cap P) \neq \emptyset$ then
10:    $T.P_P(act^{-1}) \leftarrow \bigcup_{l=1}^{|T \cap P|} p_l.P_P(act^{-1})$
11: if $(T \cap O) \neq \emptyset$ then
12:    $P_{Sys}(act) \leftarrow \bigcup_{k=1}^{|T \cap O|} P_{Sys}(act^{-1}, o_k.type)$
13: else
14:    $P_{Sys}(act) \leftarrow$ system's policy for $act$
15: (Policy Evaluation Phase)
16: for all policies in $s.P_{AS}(act)$, $T.P_{TU}(act^{-1})$, $T.P_{TS}(act^{-1})$, $T.P_O(act^{-1})$, $T.P_P(act^{-1})$ and $P_{Sys}(act)$ do
17:    Extract graph rules ($start$, $path\ rule$) from policies
18:    Get the controlling user $u_c$, if the policy is not specified by $s$ or any $t \in T$
19:    for all graph rules extracted do
20:        Determine the starting node, specified by $start$, where the path evaluation starts
21:        if graph rule is extracted from $s.P_{AS}(act)$ and $P_{Sys}(act)$ then
22:            if $start = s$ then
23:                $u_c$ and every $t \in T$ becomes the evaluating node
24:            else
25:                every $t \in T$ becomes the evaluating node
26:        else
27:            $s$ becomes the evaluating node
28:        Extract path rules $path\ rules$ from graph rules
29:        Extract each path spec $path$, $hopcount$ from path rules
30:        Path-check each path spec for each pair of starting and evaluating node
31:        Evaluate a combined result based on conjunctive or disjunctive connectives between path specs
32: Compose the final result from the result of each policy using $CRP_{Sys}$
```

D. Hopcount Skipping

According to the classic idea of "six degrees of separation" and the results of "small world experiment" [23], [29], any pair of persons are distanced by about six people on average. A recent study by Backstrom et al [1] further indicates that on the current social graph of Facebook, the average distance has shrunk to 4.74. Therefore, the network of U2U relationships is characterized by short path lengths, and the hopcount limit in a practical policy is not likely to be a large number. In contrast, U2R and R2R relationships may exhibit a different characteristic. For example, comment may be followed up by a sequence of comments, which may take a long journey for the author of the first comment to reach the author of the last comment. For this and similar cases, we introduce the "skipping" of hopcount limit of resource-related relationships, which differentiate the global hopcount limit on U2U relationships only from the possible long distance of the resource-related relationships that are found in two entities involved in request.

VI. Use Cases

Given the social graph depicted in Figure 3, below we show how access control of these examples can be realized within the model.

Example 1: Run into a new acquaintance in a photo. *Alice and Dave are strangers. Dave realizes that Alice and him both commented on Bob's photo, so he decides to poke her to say hello:*

$$(Dave, poke, Alice)$$

We need the following policies to determine authorization:

- $Dave$'s $P_{AS}(poke)$: $< poke, (u_a, ([Comment][[CommentTo\cdot CommentTo^{-1}, 2]][Comment^{-1}], 2)) >$
- $Alice$'s $P_{TU}(poke^{-1})$: $< poke^{-1}, (t, ([Comment] [[CommentTo\cdot CommentTo^{-1}, 2]][Comment^{-1}], 2)) >$
- $P_{Sys}(poke)$: $< poke, (u_a, ([\Sigma_{u\_r}][[\Sigma_{r\_r}*, 2]][\Sigma_{u\_r}], 2)) >$

The comments from Alice and Dave are connected through Bob's photo with two R2R relationships. Dave's policy says that he is free to poke his fellow commenter, while Alice allows her fellow commenter to poke her. The system facilitates many kinds of participating users (e.g., comment, like, share, etc.) to poke each other.

Example 2: View a photo where a friend is tagged. *Bob and Ed are friends of Alice, but not friends of each other. Alice posted a photo and tagged Ed on it. Later, Bob sees the activity from his news feed and decides to view the photo:*

$$(Bob, read, Photo2)$$

In this example, Bob is trying to access a resource through his friend Alice. Whether his request can be granted or not depends on the corresponding policies from himself, the target resource and the system.

- Bob's $P_{AS}(read)$: $< read, (u_a, ([\Sigma_{u\_u}*, 2][[\Sigma_{u\_r}, 1]], 2)$
- $Photo2$'s $P_O(read^{-1})$ by $Alice$: $< read^{-1}, (t, ([post^{-1}, 1][friend*, 3], 4)) >$
- $Photo2$'s $P_O(read^{-1})$ by Ed: $< read^{-1}, (u_c, ([friend], 1)) >$
- $AP_{Sys}(read)$: $< read, (u_a, ([\Sigma_{u\_u}*, 5][[\Sigma_{u\_r}, 1]], 5) >$
- $CRP_{Sys}(read)$: $< read^{-1}, (own > tag) >$

Bob, as the access requester, allows himself to read any resource that has a direct relationship with his contacts within two hops. Note that "[[]] indicates that the local hopcount "1" is not counted in the global hopcount limit "2". Alice is the original owner of the photo and Ed's image is on it, so based on our default policy, both of them are able to express their own preferences on how the photo should be exposed to others. Alice decides to share the photo with all her direct and indirect friends within three hops, while Ed prefers to keep his privacy and only wants his direct friends to see it. The system, on the other hand, specifies a more liberal rule to promote sharing that allows a user to access resource that relates to his contacts within five hops. We notice that Alice and Ed's authorization policies are apparently in conflict, which needs to be resolved. $CRP_{Sys}(read)$ says that owner's policy takes precedence over tagged user's, so the decision module will ignore Ed's policy and only consider Alice's policy. A system may configure $CRP_{Sys}(read)$ with conjunction or disjunction of the owner's and tagged users' policies for different decisions.

Example 3: Friend recommendation. *Alice is a friend of Bob, Paul follows Bob, while Alice and Paul are strangers. Bob would like to recommend Alice and Paul to be friends:*

$$(Bob, suggest\_friend, Alice, Paul)$$

Policies applied to this example are shown as follows:

- Bob's $P_{AS}(suggest\_friend)$: $< suggest\_friend, (u_a, ([\Sigma_{u\_u}*], 2)) >$
- $Alice$'s $P_{TU}(suggest\_friend^{-1})$: $< suggest\_friend^{-1}, (t, ([friend], 1)) >$
- $Paul$'s $P_{TU}(suggest\_friend^{-1})$: $< suggest\_friend^{-1}, (t, ([friend*], 2)) >$
- $P_{Sys}(suggest\_friend)$: $< suggest\_friend, (u_a, ([\Sigma*], 2)) \wedge (t, ([\Sigma*], 2)) >$

The access request contains two targets Alice and Paul, so we need target user policies from both of them. Bob can suggest friends for his contacts within two hops. Alice welcomes friend recommendation from her direct friends, while Paul allows his friends of friends to do that. The system-specified policy is more liberal, allowing users with any relationship of two hops to be able to suggest friends (e.g., two people who commented on the same photo).

Example 4: Parental control of policies. *The system features parental control such as allowing parents to configure their children's policies. The policies are used to control the incoming or outgoing activities of children, but are subject to the parents' will. For instance, Bob's mother Carol requests to set some policy, say $Policy1$ for Bob:*

$$(Carol, specify\_policy, Policy1)$$

The following policies are used to make access decision:

- $Carol$'s $P_{AS}(specify\_policy)$: $< specify\_policy, (u_a, ([own], 1) \vee ([child\cdot own], 2)) >$
- $Policy1$'s $P_P(specify\_policy^{-1})$ by Bob: $< specify\_policy^{-1}, (t, ([own^{-1}], 1) >$
- $P_{Sys}(specify\_policy)$: $< specify\_policy, (u_a, ([own], 1) \vee([child\cdot own], 2)) >$

- $CRP_{Sys}(specify\_policy)$:
  $< specify\_policy, (parent \wedge @) >$

Carol's policy offers her the ability to define her and her child's policies. Bob only allows himself to manage his own policies. The system enables parental control with the child's consent, so that parents can control their children's policies.

VII. Conclusion

In this paper, we developed an access control model for OSNs that provides finer-grained access control for users' usage and administrative access by utilizing user-to-user, user-to-resource and resource-to-resource relationship-based policies. These policies are specified in terms of relationship path patterns between the accessing user and the target together with hopcount limit of the relationships. Specifically, we introduce the skipping of some relationship path expression in the policy specification in order to offer more expressive policies. The decision modules of the system determine authorizations by retrieving different policies from the accessing session, the target and the system, and then making a collective decision. To address policy conflicts, we apply conflict resolution policies over relationship precedence. In the future, we are planning to extend our model to incorporate attribute-based controls. We also plan to extend our path checking algorithm of U2U relationships to cover the U2R and R2R relationships. Finally, we plan to undertake performance and scalability experiments with these algorithms.

Acknowledgments

This work was supported by grants from the US National Science Foundation CNS 0831452.

References

[1] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna. Four degrees of separation. *CoRR*, abs/1111.4570, 2011.

[2] S. Benferhat, R. El Baida, and F. Cuppens. A stratification-based approach for handling conflicts in access control. In *Proceedings of the eighth ACM symposium on Access control models and technologies*, SACMAT '03, pages 189–195, New York, NY, USA, 2003. ACM.

[3] E. Bertino, S. Jajodia, and P. Samarati. Supporting multiple access control policies in database systems. *ACM Transactions on Database Systems*, 26:2001, 1996.

[4] E. Bertino, S. Jajodia, and P. Samarati. A flexible authorization mechanism for relational data management systems. *ACM Trans. Inf. Syst.*, 17(2):101–140, Apr. 1999.

[5] E. Bertino, P. Samarati, and S. Jajodia. Authorizations in relational database management systems. In *Proceedings of the 1st ACM conference on Computer and communications security*, CCS '93, pages 130–139, New York, NY, USA, 1993. ACM.

[6] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. A semantic web based framework for social network access control. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, SACMAT '09, pages 177–186, New York, NY, USA, 2009. ACM.

[7] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Semantic web-based social network access control. *Computers and Security*, 30(2C3):108 – 115, 2011. Special Issue on Access Control Methods and Technologies.

[8] B. Carminati, E. Ferrari, and A. Perego. Rule-based access control for social networks. In R. Meersman, Z. Tari, and P. Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, volume 4278 of *Lecture Notes in Computer Science*, pages 1734–1744. Springer Berlin / Heidelberg, 2006.

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

[10] B. Carminati, E. Ferrari, and A. Perego. Enforcing access control in web-based social networks. *ACM Trans. Inf. Syst. Secur.*, 13(1):1–38, 2009.

[11] Y. Cheng, J. Park, and R. Sandhu. A user-to-user relationship-based access control model for online social networks. In *Proceedings of the 26th IFIP Annual WG 11.3 Conference on Data and Application Security and Privacy (DBSec '12)*, 2012.

[12] F. Cuppens, N. Cuppens-Boulahia, and M. B. Ghorbel. High level conflict management strategies in advanced access control models. *Electronic Notes in Theoretical Computer Science*, 186(0):3 – 26, 2007. Proceedings of the First Workshop in Information and Computer Security (ICS 2006).

[13] D. E. Denning, T. F. Lunt, R. R. Schell, W. R. Shockley, and M. Heckman. The seaview security model. In *Proceedings of the 1988 IEEE conference on Security and privacy*, SP'88, pages 218–233, Washington, DC, USA, 1988. IEEE Computer Society.

[14] P. Fong, M. Anwar, and Z. Zhao. A privacy preservation model for Facebook-style social network systems. In *ESORICS 2009: Proceedings of 14th European Symposium on Research in Computer Security, Saint-Malo, France, September 21-23, 2009*, page 303. Springer, 2009.

[15] P. W. Fong. Relationship-based access control: protection model and policy language. In *Proceedings of the first ACM conference on Data and application security and privacy*, CODASPY '11, pages 191–202, New York, NY, USA, 2011. ACM.

[16] P. W. Fong and I. Siahaan. Relationship-based access control policies and their policy languages. In *Proceedings of the 16th ACM symposium on Access control models and technologies*, SACMAT '11, pages 51–60, New York, NY, USA, 2011. ACM.

[17] C. E. Gates. Access control requirements for web 2.0 security and privacy. In *Proc. of Workshop on Web 2.0 Security and Privacy (W2SP 2007)*, 2007.

[18] H. Hu and G.-J. Ahn. Multiparty authorization framework for data sharing in online social networks. In *Proceedings of the 25th annual IFIP WG 11.3 conference on Data and applications security and privacy*, DBSec'11, pages 29–43, Berlin, Heidelberg, 2011. Springer-Verlag.

[19] S. Jajodia, P. Samarati, and V. S. Subrahmanian. A logical language for expressing authorizations. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 31–42, 1997.

[20] S. Jajodia, P. Samarati, V. S. Subrahmanian, and E. Bertino. A unified framework for enforcing multiple access control policies. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, SIGMOD '97, pages 474–485, New York, NY, USA, 1997. ACM.

[21] S. Kruk, S. Grzonkowski, A. Gzella, T. Woroniecki, and H. Choi. D-FOAF: Distributed identity management with access rights delegation. *Lecture Notes in Computer Science*, 4185:140, 2006.

[22] A. Masoumzadeh and J. Joshi. Osnac: An ontology-based access control model for social networking systems. In *IEEE Social Computing (SocialCom)*, 2010.

[23] S. Milgram. The small world problem. *Psychology Today*, 2(1):60–67, 1967.

[24] J. Park, R. Sandhu, and Y. Cheng. Acon: Activity-centric access control for social computing. In *Proceedings 5th International Conference on Availability, Reliability and Security (ARES)*, 2011.

[25] J. Park, R. Sandhu, and Y. Cheng. A user-activity-centric framework for access control in online social networks. *Internet Computing, IEEE*, 15(5):62–65, sept.-oct. 2011.

[26] F. Rabitti, E. Bertino, W. Kim, and D. Woelk. A model of authorization for next-generation database systems. *ACM Trans. Database Syst.*, 16(1):88–131, Mar. 1991.

[27] H. Shen and P. Dewan. Access control for collaborative environments. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, CSCW '92, pages 51–58, New York, NY, USA, 1992. ACM.

[28] A. C. Squicciarini, M. Shehab, and F. Paci. Collective privacy management in social networks. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 521–530, New York, NY, USA, 2009. ACM.

[29] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):pp. 425–443, 1969.