A New Access Control Scheme for Facebook-style Social Networks

Jun Pang[a,b,*], Yang Zhang[a]

[a] *University of Luxembourg, Faculty of Sciences, Technology and Communication*
6, rue Coudenhove-Kalergi, L-1359 Luxembourg
[b] *University of Luxembourg, Interdisciplinary Centre for Security, Reliability and Trust*
4, rue Alphonse Weicker, L-2721 Luxembourg

Abstract

The popularity of online social networks (OSNs) makes the protection of users' private information an important but scientifically challenging problem. In the literature, relationship-based access control schemes have been proposed to address this problem. However, with the dynamic developments of OSNs, we identify new access control requirements which cannot be fully captured by the current schemes. In this paper, we focus on public information in OSNs and treat it as a new dimension which users can use to regulate access to their resources. We define a new OSN model containing users and their relationships as well as public information. Based on this model, we introduce a variant of hybrid logic for formulating access control policies. We exploit a type of category information and relationship hierarchy to further extend our logic for its usage in practice. In the end, we propose a few solutions to address the problem of information reliability in OSNs, and formally model collaborative access control in our access control scheme.

Key words: Social networks, access control, privacy, hybrid logic.

1. Introduction

Online social networks (OSNs) are among the most popular web services during the past ten years and have attracted a huge amount of users all over the world. For example, Facebook, the leading OSN service, has more than one billion active users monthly.[2] OSNs are playing an important role in our daily life by providing a platform for users to present themselves, articulate their social circles, interact with each other etc.

[*] Corresponding author. Tel: +352 466 644 5483, fax: +352 466 644 35483

Email addresses: `jun.pang@uni.lu` (Jun Pang), `yang.zhang@uni.lu` (Yang Zhang)

This article is a revised and extended version of [1] that has appeared in the proceedings of the 9th IEEE Conference on Availability, Reliability and Security (ARES 2014).

[2] `http://newsroom.fb.com/`

With the large amount of data maintained in OSN websites, privacy concerning users' personal information inevitably becomes an important but scientifically challenging problem. Access control schemes (e.g., see [2, 3, 4, 5, 6, 7, 8, 9]) are naturally introduced to protect users' private information or resources in OSNs. They can be used to guarantee that resources are only accessible by the intended users, but not by other (possibly malicious) users. Users can control the access to their own information or resources with access control schemes supplied by OSNs. The existing schemes, including the ones proposed by the research community, are mainly *relationship-based*, i.e., whether a user is able to access the information depends on the relationship between him and the owner, e.g., 'friends' or 'friends of friends'.

Due to their own nature and the development of information and communications technology, OSNs admit quick and dynamic evolutions. Many new services and methods for user interaction have emerged. For instance, users can play online games with friends or find people who share similar interests. More recently, with the increased popularity of GPS-enabled mobile devices, OSNs have evolved into geo-social networks – users can tag posts and photos with their geographical locations, find nearby friends and post check-in of some places to share their comments. OSNs are also emerging as important social media – people use OSNs to publish news, organize events or even seek for emergent help. For example, Facebook and Twitter play an extremely important role during the rescue process for the "April 2011 Fukushima earthquake"; and in summer 2014, the "Ice Bucket Challenge" have achieved a huge success through social media.[3] (In Section 3, we will take Facebook as a typical example and discuss its developments in the past few years.)

With these evolutions, more information and activities of users are made available in OSNs. As a result, new access control schemes are needed to capture these new developments. Let us illustrate this need by a few scenarios in OSNs.

- Someone broke the window of Alice's expensive car and took her purse when she parked the car in the area of Montparnasse in Paris. Alice publishes a status in the OSN to see if anyone can provide her some clue to find the purse back. She doesn't want everyone to know that she has an expensive car, and people who live in other areas or cities won't be able to give her any useful information. Therefore, she intends to choose people who live in the Montparnasse area as audiences of her status.

- Bob wants to organize a fundraising party for children's rare diseases. He doesn't want to make this event public as certain sensitive information of the participants can be leaked, e.g., it is possible that some participants' family members may suffer from the disease. Instead, Bob only wants people who are linked with a certain number of charities (through donations, volunteering, etc) as him to attend the party.

[3] `http://en.wikipedia.org/wiki/Ice_Bucket_Challenge`

- Charlie has some friends who work at the rival company of his own employer. These friends invited him to attend the party organized by their company. Charlie publishes a photo taken at the party. Apparently, it is not a good idea for his colleagues and boss to see this photo. Thus Charlie wants no one but his friends who work at this rival company to see it.

In relationship-based schemes, a resource owner cannot exploit any other information but user relationships between him and the requester when defining access control policies. Therefore, the above requirements cannot be fully and precisely formulated in the current schemes proposed in the literature.

Contributions and Outline. In order to solve the identified problems, we propose a new access control scheme for OSNs. We focus on public information existing, e.g., in Facebook (Section 3), and show that it can be used to group users based on their attributes, common interests and activities. Public information can thus be considered as a new dimension for users to regulate access to their resources. As a consequence, we propose a new OSN model containing both a user graph and a public information graph (Section 4). We then extend a hybrid logic [10] to express this type of access control policies (Section 5). The expressiveness of our scheme is extensively discussed through a number of real-life scenarios (Section 6). We further identify two special semantic relations, i.e., *category relation* among public information and *relationship hierarchy*, which allow us to express certain types of policies in a concise way (Section 7 and Section 8). To address the problem of information reliability in OSNs, we propose to add endorsement and trust into our policy formulas (Section 9). In addition, we formally model the collaborative access control in Section 10 within our new access control scheme.

After the introduction, we give a brief overview of related work in Section 2. Section 11 compares our access control scheme with existing schemes in the literature. We discuss several issues related to our scheme in Section 12 and conclude our paper with some future work in Section 13.

2. Related Work

Relationship-based access control, driven by OSNs, was first advocated in [11] and defined as an access control paradigm based on interpersonal relationships. Carminati et al. proposed the first relationship-based access control model in [12], where the relationships between the qualified requester and the owner are interpreted into three aspects, i.e., relationship type, depth and trust level. In [13], the authors used semantic web technology including OWL and SWRL to extend the model of [12]. They also proposed administrative and filtering policies which can be used for collaborative and supervising access control, respectively. Fong et al. proposed an access control scheme for Facebook-style social networks [14], in which they model the access control procedure as two stages. In the first stage, the requester has to find the owner of the target resource; then in the second stage, the owner decides whether the authorization is granted or not. Their access control policies are mainly based on the relationships between the requester and the owner. Moreover, they proposed several meaningful access control policies based on the graph structure of OSNs, such as n-common friends and clique. In [15], Fong introduced a modal logic to define access control policies for OSNs. Later Fong and Siahaan [16] improved the previously proposed logic to further support policies like n-common friends and clique. In [10], the authors adopted a hybrid logic to describe policies which eliminates an exponential penalty in expressing complex relationships such as n-common friends. This hybrid logic is expressive and has been adopted by several other works [17, 18, 19] for specifying access control policies. A visualization tool for evaluating the effect of access control configurations is designed in [20], with which a user can check which other users within a certain distance to him can view his resources. Cheng et al. proposed a rich OSN model in [21]. In their work, not only users but also resources are treated as entities and actions performed by users are considered as relationships in OSNs. As more information are incorporated in their model, many new access control policies can be expressed (more details can be found in Section 11). Their model supports administrative and filtering policies as proposed in [13]. Recently, Crampton and Sellwood [22] generalized relationship-based access control to other systems than social networks, they proposed path logic conditions for specifying policies and adopt principle matching for policy evaluation. Besides models, several security protocols based on cryptographic techniques are proposed to enforce relationship-based access control policies, e.g., see [23, 24, 25, 26, 27, 28, 29, 30, 31].

As a shared platform, resources in OSNs may be co-owned by a number of users. Thus, collaborative access control also plays an essential role in protecting privacy. A game theoretical method based on the Clarke-Tax mechanism for collective privacy management was proposed by Siquicciarini et al. [32]. Sun et al. proposed a different approach by combining trust relations in OSNs and preferential voting schemes [33, 34]. Ahn et al. introduced a multiparty access control model in [35]. In addition, they developed a policy specification scheme and a voting based conflict resolution mechanism. Photo tagging is the most common service relevant to collaborative access control. The authors of [36, 37] have investigated users' privacy concerns about this service and proposed principles for designing better collaborative access control schemes. Besides interaction, users' private information can be leaked through third party applications. A privacy-by-proxy design for social network APIs was developed by Felt and Evans [38]. Singh et al. [39] proposed a privacy-preserved application platform, i.e., `xBook`, which integrates information flow model to control what applications can do with users' information. An access control scheme for third party applications was developed [40], where applications are required to adapt users' specifications on their own data.

3. Motivation

An OSN provides users with some typical services, such as users can build their profiles and establish social relationships with each other. Moreover, an OSN also provides a platform for users to socialize and interact with each other.

In the following, we first give a brief overview of the developments of Facebook – one of the most popular OSN services in the world. After that, we discuss public information in Facebook and its potential usage in access control.

3.1. Facebook

In Facebook, each user is affiliated with a personal profile that contains his basic information (e.g., age, gender and nationality), work and education background, living places and so on. A user's hobbies (e.g., sports, movies and music) are articulated in 'Likes'; places he has been to are marked in 'Map'. Besides personal representation, Facebook also allows a user to establish friend relations with others. In addition, Facebook recommends friends for users based on common friends, same hometown or similar interests. A user can organize his friends into different groups, or named friend list. Moreover, Facebook also automatically create lists (smart list) for users based on their work, living area, school and family.

Facebook is not only a website storing users' personal information and social relations, but also a platform for users to interact with each other. A user can directly communicate with his friends by sending messages; he can tag his friends in photos and posts. Two friends can interact through Facebook applications such as games. All activities performed by a user are organized chronologically in his 'Timeline' through which other users as well as the user himself can check his past activities conveniently. A user receives his friends' news on 'Newsfeed'. When he finds something interesting, he can further perform actions, such as 'like', 'share' and 'comment', on it. Users on Facebook can also establish public groups and organize events such as birthday parties, meetings and conferences, and invite other users to attend, like or share these groups or events.

In January 2013, Facebook publishes a new product called Graph Search, a search engine based on users' data.[4] It allows users to explore more information about daily life, find people who share common interests or live in the same city, discover new restaurants and music, and so on. Through Graph Search, a user can directly acquire information from his friends' data without visiting their personal pages. For example, if a user types in "photos by my friends", he will get a page containing all photos uploaded by his friends. Since Graph Search is a personalized search engine, for the same query different users will get different results.

When a user wants to publish or share a resource (a photo or a post), Facebook provides him an audience selector to let him decide who can view this resource. This audience selector is the access control implementation of Facebook and it supports five different modes including 'public', 'friends', 'friends except acquaintances', 'only me' and 'custom'. In the last mode, a user can choose the eligible requester to be (or not to be) a single user or a specific group (through friend list). The selector also supports smart lists in Facebook, which group a user's friends according to 'work', 'school', 'family' and 'city' using the

[4] `https://www.facebook.com/about/graphsearch`

information added to the 'Education' and 'Work' and 'Current City' sections of their profile.

3.2. Public Information and Access Control

Besides users' information, Facebook imports knowledge of external sources, e.g., Wikipedia and Bing map, into its system to formalize another type of entities. We name them *public information.* A lot of entities in the real world are modeled as public information, e.g., countries, history events or public figures. Public information are mainly used as common reference points of users' information, through which a user can find other users in Facebook with similar background, hobbies, experiences, etc. For example, a user can find his schoolmates through the public information of the college that he has attended.

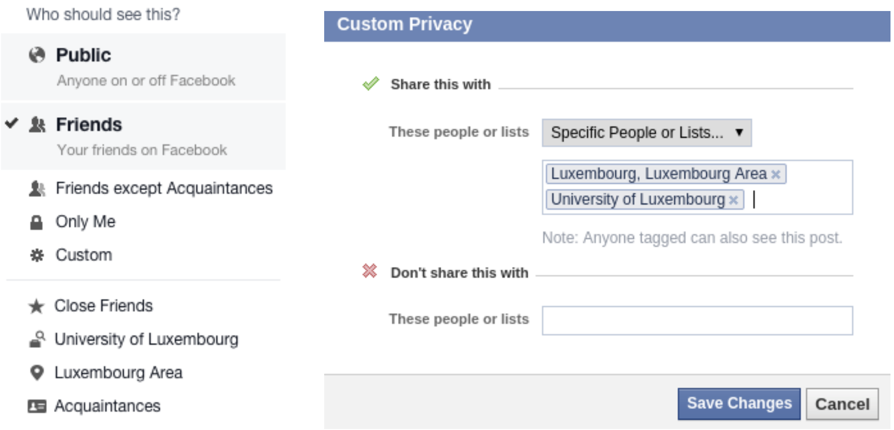
Each public information is affiliated with a content that is normally extracted from external sources. Similar to users, public information are also connected with each other and links among them are based on their contents. For example, if Wikipedia articles of two charities are connected, then their public information in Facebook are connected as well. Besides, there exist many different links between users and public information. Some of these connections are based on user profiles, e.g., if a user specifies his employer in his profile, then he is linked with this employer's public information. Others are computed by Facebook through mining users' data. For example, if a user posts a status labeled with a location, then the user is connected with the location's public information.

In addition to facilitate users' interaction, it is possible to use public information in expressing access control requirements. For example, in the first scenario as discussed in Section 1, the requester has to be linked to the location where the car was parked; in the second one, the requester needs to be linked with the owner through some charity organizations; in the third one, the requester is asked to be connected with the owner through not only a friendship but also their employers' connection. Here, the location, charities as well as companies can all be modeled as public information in OSNs.

All the above access control requirements are meaningful and in line with the recent developments of OSNs. However, the current access control schemes proposed in the literature mainly focus on relationships among users, public information are not taken into account. On the other hand, Facebook already allows users to define policies with some simple public information. As shown in Figure 1, a user can define a policy to allow users who lives in the same area or work at the same university as him to view his photo through smart lists. However, this function is still ad hoc, scenarios proposed in Section 1 cannot be fully captured. Therefore, in this paper we propose a new access control scheme, in which policies can be expressed based on both users and public information, and their relationships.

4. A Model of Online Social Networks

Our OSN model contains information of (1) users and their social relationships, (2) public information and their connections, and (3) links between users



✓ Share this with

These people or lists: Specific People or Lists... ▼

Luxembourg, Luxembourg Area ×
University of Luxembourg ×

Note: Anyone tagged can also see this post.

✗ Don't share this with

These people or lists:

Figure 1: Access control with smart list in Facebook.

and public information. Public information and users are essentially two different concepts – public information are imported from external databases (in most cases), and they cannot perform actions and establish relationships with each other as users; relationships among public information are also extracted from external sources. Therefore, we treat public information and users separately. We model an OSN as a tuple $(\mathcal{UG}, \mathcal{PG}, \rho, \varrho)$. A user graph is denoted by $\mathcal{UG}$, and it depicts users and their relationships. A public information graph is denoted by $\mathcal{PG}$, which represents all public information and connections among them. Two maps, i.e., ρ and ϱ, store links between users and public information.

4.1. User Graph

The set $\mathcal{U}$ contains all users in an OSN. Each user is affiliated with some basic information which are treated as attributes of the user. We use $\mathcal{UR} = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ to denote a (finite) set of relationship types supported in the OSN. The semantics of each relationship type is defined as $\alpha_i \subseteq \mathcal{U} \times \mathcal{U}$. If user u_a is in a α_i relationship with user u_b, then we write $(u_a, u_b) \in \alpha_i$. For each relationship type $\alpha_i \in \mathcal{UR}$, there exists its reverse relationship type, e.g., if α_i stands for *husbandof*, then its reverse is *wifeof*. We use $\alpha_i^{-1} \in \mathcal{UR}$ to denote the reverse of α_i. Moreover, if $\alpha_i = \alpha_i^{-1}$, then α_i is a *symmetric relationship*, e.g., *friend* is a typical symmetric relationship. User graph $\mathcal{UG}$ is a directed graph denoted as $(\mathcal{U}, \mathcal{UE})$, where every user in the OSN is a node and the set of edges, i.e., $\mathcal{UE}$, is defined as $\{(u_a, u_b, \alpha_i) \mid u_a, u_b \in \mathcal{U}$ and $(u_a, u_b) \in \alpha_i\}$.

4.2. Public Information Graph

As we introduced in Section 3, public information are also linked as together, such as Paris is linked with France. Therefore, we model public information as a graph. We use the set $\mathcal{P}$ to denote all public information that are extracted from external databases, such as Wikipedia and some geography

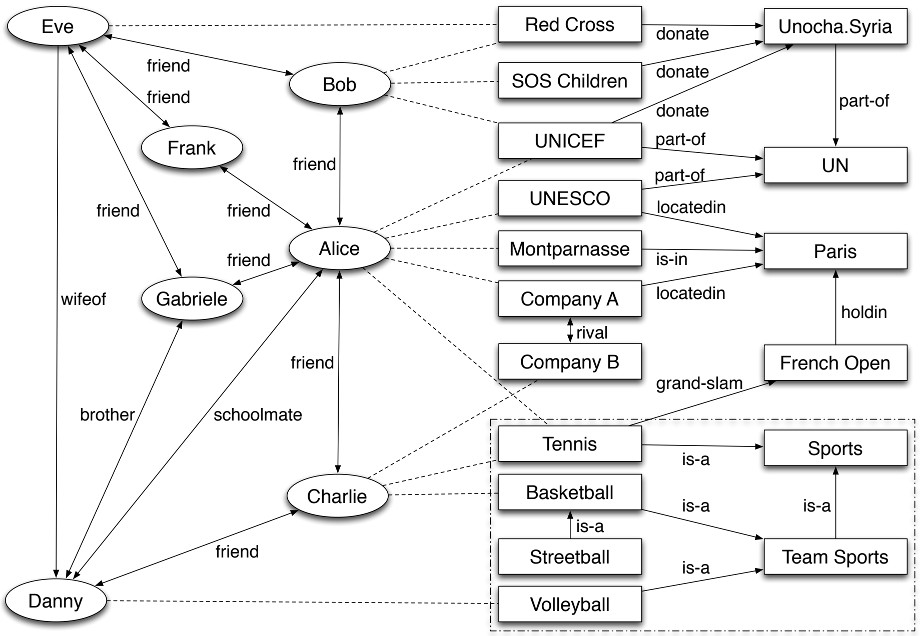


Figure 2: A sample OSN model.

databases (such as Bing). Each public information f_c has its own attributes. We use $\mathcal{PR} = \{\beta_1, \beta_2, \ldots, \beta_\ell\}$ to denote a (finite) set of relationship types on public information. Each relationship type β_j can be semantically defined as $\beta_j \subseteq \mathcal{P} \times \mathcal{P}$. If β_j's reverse relationship type exists, it is denoted by β_j^{-1}. Public information graph is formally denoted as $\mathcal{PG} = (\mathcal{P}, \mathcal{PE})$, where $\mathcal{P}$ is the set of nodes and $\mathcal{PE}$ is defined as $\{(f_c, f_d, \beta_j) \mid f_c, f_d \in \mathcal{P} \text{ and } (f_c, f_d) \in \beta_j\}$.

4.3. Links between $\mathcal{UG}$ and $\mathcal{PG}$

There are a lot of links between users and public information. For example, a user is linked with the language he speaks and the city he lives in. As the OSN is modeled as $\mathcal{UG}$ and $\mathcal{PG}$, we define two maps, i.e., ρ and ϱ, between them to describe their connections:

$$\rho: \quad \mathcal{U} \quad \to \quad 2^{\mathcal{P}} \qquad \text{and} \qquad \varrho: \quad \mathcal{P} \quad \to \quad 2^{\mathcal{U}}.$$

For a user $u_a \in \mathcal{U}$, $\rho(u_a)$ is a subset of the nodes in $\mathcal{PG}$ that are related to u_a. The map $\rho(u_a)$ may contain a lot of different types of public information, such as museums, universities, pop stars, etc, which are computed by the OSN with the information that u_a provides. For a public information $f_c \in \mathcal{P}$, $\varrho(f_c)$ gives all the users in $\mathcal{UG}$ who have been involved in activities or have information related to f_c. How to compute ρ and ϱ is not the focus of this paper, we assume that ρ and ϱ always give us the right results. In practice, it is desirable to have more fine-grained links between users and public information. With respect to this, the two maps ρ and ϱ can be further refined to reflect how precisely a user and a piece of public information is connected.

4.4. An Example

A sample OSN model is shown in Figure 2, whose left side is a $\mathcal{UG}$ and right side is a $\mathcal{PG}$. Edges in the graph with double arrows imply that the relationships are symmetric.[5] For example, Alice and Bob are friends; Company A and Company B are rivals. The dash lines between users and public information reflect the links between $\mathcal{UG}$ and $\mathcal{PG}$, which are formally captured by the two maps ρ and ϱ. (The part contained in the dashed box in the right-bottom corner will be discussed in Section 7.)

5. A Hybrid Logic

In [10], a hybrid logic is used to define access control policies for OSNs. We adopt their logic and additionally introduce a new type of formulas ψ. With such formulas, we can define policies based on information in $\mathcal{PG}$. Moreover, two new logic operators, i.e., $\triangleright$ and $\blacktriangleright$, are introduced to connect formulas on $\mathcal{UG}$ and $\mathcal{PG}$, respectively. In this way, we can combine resources and their relations from both $\mathcal{UG}$ and $\mathcal{PG}$ to specify new and expressive access control policies (see examples in Section 6).

5.1. Syntax

The syntax of our hybrid logic is given below, and its semantics will be discussed in the next section.

$$\begin{array}{rcl}
s & ::= & m \mid x \\
t & ::= & n \mid y \\
\phi & ::= & s \mid p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \langle\alpha_i\rangle\phi \mid \bigcirc_s\phi \mid \nabla_x\phi \mid \triangleright\psi \\
\psi & ::= & t \mid q \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \langle\beta_j\rangle\psi \mid \bullet_t\psi \mid \blacktriangledown_y\psi \mid \blacktriangleright\phi
\end{array}$$

In our logic, there are mainly two types of formulas: the user formulas ϕ manipulate information on the user graph $\mathcal{UG}$, while the public information formulas ψ are defined on $\mathcal{PG}$. Three kinds of atoms are supported in our logic, i.e., nominals (m and n), variables (x and y) and proposition symbols (p and q). Nominal m represents the name of a user in $\mathcal{UG}$, e.g., Alice, while n represents the name of a public information in $\mathcal{PG}$, e.g., Paris. Propositional symbol p is used for specifying the attributes of users in $\mathcal{U}$ and similarly q is used for public information in $\mathcal{P}$. For example, p (i.e., *IsMale*) can specify users who are male and q (i.e., *IsCity*) can specify those publication information representing a city. Atoms m, x and p are used in user formulas ϕ, while n, y and q are used in

[5] For the sake of simplicity, we omit some edges in the figure, e.g., the edge from Danny and Eve to represent the relationship 'husbandof'.

public information formulas ψ. Negation $\neg$ and conjunction $\wedge$ have their usual meanings and can be used to define disjunction $\vee$. Therefore, we also use $\vee$ in both ϕ and ψ. $\langle\alpha_i\rangle$ and $\langle\beta_j\rangle$ are two modal logic operators. As described in Section 4, symbols α_i and β_j represent the relationship types in $\mathcal{UG}$ and $\mathcal{PG}$, respectively. Hybrid logic operator $\bigcirc$ can be used either with a nominal or variable, while $\triangledown$ can only operate on variables. The same holds for $\CIRCLE$ and $\blacktriangledown$. Two new logic operators, i.e., $\triangleright$ and $\blacktriangleright$, are used to connect the two types of formulas ϕ and ψ together. They allow the specification of access control policies based on both information from the user graph and the public information graph.

5.2. Semantics

Our model for evaluating access control policy formulas contains six parts, i.e., $\Gamma, \Delta, \rho, \varrho, \mathit{cur\_n}, \tau$, where $\Gamma = (\mathcal{UG}, V_U)$ and $\Delta = (\mathcal{PG}, V_P)$. V_U is a map between atoms (either m or p) and users in $\mathcal{UG}$, $V_U(m)$ is a set that contains only one user in $\mathcal{UG}$ whose name is m and $V_U(p)$ is a set of users that have the attribute as specified by p. For example, $V_U(\mathit{Alice})$ refers to a singleton containing the node of Alice in $\mathcal{UG}$. Similarly, we can define $V_P(n)$ and $V_P(q)$. As introduced in Section 4, ρ and ϱ maintained by the OSN connect users and public information. Node $\mathit{cur\_n}$ refers to either a user u_a in $\mathcal{UG}$ or a public information f_c in $\mathcal{PG}$. Valuation τ stores all the maps from variables in the policy formula to vertices in either $\mathcal{UG}$ or $\mathcal{PG}$. When there is a new map from x to u_a (y to f_c) added to τ, we write $\tau[x \mapsto u_a]$ ($\tau[y \mapsto f_c]$).

We use satisfaction relation $\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash \phi$ to describe the meaning of user formula ϕ.

$$\begin{array}{ll}
\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash x & \text{iff } u_a = \tau(x) \\
\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash m & \text{iff } V_U(m) = \{u_a\} \\
\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash p & \text{iff } u_a \in V_U(p) \\
\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash \neg\phi & \text{iff } \Gamma, \Delta, \rho, \varrho, u_a, \tau \nvDash \phi \\
\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash \phi_1 \wedge \phi_2 & \text{iff } \Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash \phi_1 \wedge \Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash \phi_2 \\
\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash \langle\alpha_i\rangle\phi & \text{iff } \exists u_b \in \mathcal{U} \text{ s.t. } (u_a, u_b) \in \alpha_i \wedge \Gamma, \Delta, \rho, \varrho, u_b, \tau \vDash \phi \\
\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash \bigcirc_m\phi & \text{iff } \Gamma, \Delta, \rho, \varrho, u_b, \tau \vDash \phi \text{ where } V_U(m) = \{u_b\} \\
\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash \bigcirc_x\phi & \text{iff } \Gamma, \Delta, \rho, \varrho, \tau(x), \tau \vDash \phi \\
\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash \triangledown_x\phi & \text{iff } \Gamma, \Delta, \rho, \varrho, u_a, \tau[x \mapsto u_a] \vDash \phi \\
\Gamma, \Delta, \rho, \varrho, u_a, \tau \vDash \triangleright\psi & \text{iff } \exists f_c \in \rho(u_a) \text{ s.t. } \Gamma, \Delta, \rho, \varrho, f_c, \tau \vDash \psi
\end{array}$$

The first three relations express the meaning of atoms. When ϕ is a single variable x, it holds if and only if when τ contains a map from x to u_a. If ϕ is a single nominal or propositional symbol, it is true if and only if when u_a is in the set defined by V_U. When several modal logic operators ($\langle\alpha_i\rangle$) are aligned sequentially in a formula, they can represent a *relationship path*, e.g., user can define a policy to regulate that only 'friends of friends' can access his resource.

The hybrid logic operator $\bigcirc_s\phi$ jumps to the node that s refers to in $\mathcal{UG}$, and $\triangledown_x\phi$ adds a map from x to u_a into τ. The new operator, i.e., $\triangleright\psi$, links a user formula ϕ with a public information formula ψ – it maps the current node u_a in $\mathcal{UG}$ to a set of public information in $\mathcal{PG}$ that are related to this user. If there is one public information in $\rho(u_a)$ satisfying ψ, then the formula $\triangleright\psi$ holds.

In the following, we give the meaning of public information formulas ψ.

$$
\begin{array}{ll}
\Gamma, \Delta, \rho, \varrho, f_c, \tau \models y & \text{iff } f_c = \tau(y) \\
\Gamma, \Delta, \rho, \varrho, f_c, \tau \models n & \text{iff } V_P(n) = \{f_c\} \\
\Gamma, \Delta, \rho, \varrho, f_c, \tau \models q & \text{iff } f_c \in V_P(q) \\
\Gamma, \Delta, \rho, \varrho, f_c, \tau \models \neg\psi & \text{iff } \Gamma, \Delta, \rho, \varrho, f_c, \tau \not\models \psi \\
\Gamma, \Delta, \rho, \varrho, f_c, \tau \models \psi_1 \wedge \psi_2 & \text{iff } \Gamma, \Delta, \rho, \varrho, f_c, \tau \models \psi_1 \wedge \Gamma, \Delta, \rho, \varrho, f_c, \tau \models \psi_2 \\
\Gamma, \Delta, \rho, \varrho, f_c, \tau \models \langle\beta_j\rangle\psi & \text{iff } \exists f_d \in \mathcal{P}\text{s.t.}(f_c, f_d) \in \beta_j \wedge \Gamma, \Delta, \rho, \varrho, f_d, \tau \models \psi \\
\Gamma, \Delta, \rho, \varrho, f_c, \tau \models \bullet_n\psi & \text{iff } \Gamma, \Delta, \rho, \varrho, f_d, \tau \models \psi \text{ where } V_P(n) = \{f_d\} \\
\Gamma, \Delta, \rho, \varrho, f_c, \tau \models \bullet_y\psi & \text{iff } \Gamma, \Delta, \rho, \varrho, \tau(y), \tau \models \psi \\
\Gamma, \Delta, \rho, \varrho, f_c, \tau \models \blacktriangledown_y\psi & \text{iff } \Gamma, \Delta, \rho, \varrho, f_c, \tau[y \mapsto f_c] \models \psi \\
\Gamma, \Delta, \rho, \varrho, f_c, \tau \models \blacktriangleright\phi & \text{iff } \exists u_a \in \varrho(f_c) \text{ s.t.} \Gamma, \Delta, \rho, \varrho, u_a, \tau \models \phi
\end{array}
$$

It is easy to find that the semantics of public information formulas resembles the user formulas. Therefore, information in $\mathcal{PG}$ can be used in access control policies in a same way as in $\mathcal{UG}$. When the evaluation process encounters the operator $\blacktriangleright\phi$, the public information node f_c is mapped to users that are related to it in $\mathcal{UG}$. If ϕ holds at one of these users, then the formula $\blacktriangleright\phi$ is true.

Note that, by combing the user formula $\rhd\psi$ with propositions, we can link a user to a more specific set of public information. We write $\rhd_q\psi$ for $\rhd(q \wedge \psi)$ and its meaning can be reinterpreted as:

$$\Gamma, \Delta, \rho, \varrho, u_a, \tau \models \rhd_q\psi \text{ iff } \exists f_c \in \rho(u_a) \cap V_P(q) \text{ s.t. } \Gamma, \Delta, \rho, \varrho, f_c, \tau \models \psi$$

Similarly, we can define $\blacktriangleright_p\phi$ as $\blacktriangleright(p \wedge \phi)$ and formulate its semantics.

5.3. Expressing Access Control Policies

In general, there are four elements in an access control scenario, i.e., a requester, a target, an action and access control policies. More precisely, the requester tries to perform an action on the target, whether he succeeds or not depends on the access control policies defined for the target.

- **Owner and requester.** Both the owner and requester are users in the social network, and we use free variables own and req to represent the owner of the resource and the requester in the formula.

- **Target.** With multiple services supported by the OSN, a target can be a user or a resource. For example, a requester can request to chat with a user or view one of his photo. For the sake of simplicity, we assume that the target can only be a resource owned by some user.

- **Access control policies.** Normally, a user can define an access control policy for the resources that he owns. But in some cases, the access of a resource is decided by several users. For example, for a photo that is tagged with several users, each of them should have the right to decide who can view this photo. This is the subject of collaborative (or multiparty) access control management, e.g., see [21, 32, 33, 34, 35]. For now, we assume that a resource is attached with only one access control policy that is defined by its owner. In Section 10, we will show how to support collaborative access control within our access control scheme.

- **Action.** As introduced in Section 3, a user can perform multiple actions in OSNs, such as 'view', 'comment', 'tag' and 'share'. The only action we consider here is 'view' and other actions are affiliated with it, i.e., when a user is able to view a resource published by another user, he can comment or share it as well.

In OSNs, both the requester and the owner are users. We restrict that an access control formula has to start with either an owner or a requester, i.e., policy formulas are in the form either $\bigcirc_{\mathsf{own}}\phi$ or $\bigcirc_{\mathsf{req}}\phi$.

5.4. Model Checking

Given an OSN model $(\mathcal{UG}, \mathcal{PG}, \rho, \varrho)$ and an access control policy expressed in our hybrid logic as a formula ϕ, the satisfaction of $\mathit{\Gamma}, \mathit{\Delta}, \rho, \varrho, u_a, \tau \vDash \phi$ with $\tau[\mathsf{own} \mapsto u_a, \mathsf{req} \mapsto u_b]$, $\mathit{\Gamma} = (\mathcal{UG}, V_U)$ and $\mathit{\Delta} = (\mathcal{PG}, V_P)$ is formulated as a local model checking problem by Bruns et al. [10]. Except for the user graph $\mathcal{UG}$, our OSN model captures public information and their relationships. Moreover, our logic essentially extends the one of [10] with public information formulas ψ defined on $\mathcal{PG}$ and two new operators $\triangleright$ and $\blacktriangleright$ connecting user formulas and public information formulas. In principle, we can reuse the model checking algorithm of Bruns et al. [10]. As formulas of the form $\triangleright\psi'$ or $\blacktriangleright\phi'$ explore the links between $\mathcal{UG}$ and $\mathcal{PG}$, we need to treat them differently. A formula $\triangleright\psi'$ maps the current node (cur_n) in $\mathcal{UG}$ to a a set of public information in $\mathcal{PG}$. As long as there is one public information in $\rho(cur\_n)$ satisfying ψ, then ϕ holds. The formula $\blacktriangleright\phi'$ is defined similarly. To check them, we can develop a sub-routine similar to `MCmay` of Bruns et al. [10], which first computes the set of all public information (users) related to a specific user (public information) and then iterate through the set until one of them makes the connected formula ψ' (ϕ') hold on $\mathcal{PG}$ ($\mathcal{UG}$). For formulas $\triangleright(q \wedge \psi')$ and $\blacktriangleright(p \wedge \phi')$ as discussed in Section 5, we can further reduce the size of the computed set by using propositions p and q to improve the efficiency in model checking.

6. Example Policies

In order to show the expressiveness of our new scheme based on the OSN model, we design several real-life scenarios and give their corresponding formulas in our logic. We use the OSN model depicted in Figure 2, and we assume that valuation g contains two maps $\mathsf{own} \mapsto u_o$ and $\mathsf{req} \mapsto u_r$, where $u_o, u_r \in \mathcal{U}$ are the owner and the requester, respectively.

Scenario 0. We first show how to express the policy related to user relationships. Suppose that Eve defines a policy on a certain resource to regulate that the qualified requesters can only be her friends or friends of friends. The policy formula can be written as follows:

$$\bigcirc_{\mathsf{own}}(\langle \mathit{friend} \rangle \mathsf{req} \vee \langle \mathit{friend} \rangle \langle \mathit{friend} \rangle \mathsf{req}).$$

The hybrid logic operator $\bigcirc_{\mathsf{own}}$ drives the formula to start at Eve. The requirement "friends of friends" is achieved by aligning $\langle \mathit{friend} \rangle$ twice which forms a relationship path of length two. In Figure 2, Bob, Frank and Gabriele can view the resource because they are friends of Eve, Alice is also eligible since she is one of Eve's friends of friends.

To restrict the access to the photo, except for her friends, Eve regulates that the qualified requester should have at least three common friends with her. The policy formula is written as

$$\bigcirc_{\mathsf{own}}(\langle \mathit{friend} \rangle \mathsf{req} \vee \langle \mathit{friend} \rangle_3 \mathsf{req}).$$

This is the 'n-common friends' – one of the topology-based access control policies defined in [14] – $\langle \mathit{friend} \rangle_3$ expresses 'at least three different friends' in the formula. In [10], the authors show how to implement this policy with the logic operators $\triangledown_x$ and $\bigcirc_s$, we omit the details here. In Figure 2, as Alice has three common friends with Eve, she can still view the photo.

Next, we illustrate the usage of public information by defining access control policies for four different scenarios. In the first scenario, public information are used to describe an attribute of the qualified requester. While in the second and third scenarios, the owner and the requester are linked through public information. In addition, the third scenario needs the owner and the requester to be connected through the user relationship as well. In the fourth scenario (not discussed in Section 1), the owner and the requester are linked through a path composed by both users and public information.

Scenario 1. Let us recall the first access control scenario discussed in Section 1, which exploits the information in $\mathcal{PG}$. Alice publishes a status to find a witness who lives in or visited the area where her car was broken into, i.e., Montparnasse in Figure 2. The policy is formulated as

$$\bigcirc_{\mathsf{req}} \rhd \mathsf{Montparnasse}.$$

The operator $\rhd$ links $\mathcal{UG}$ with $\mathcal{PG}$, as introduced in Section 5, we can use $\rhd_{IsLocation}$ to make the map more precisely. $\mathsf{Montparnasse}$ in the formula is a nominal, $V_P(\mathsf{Montparnasse})$ is the node that represents Montparnasse in $\mathcal{PG}$. Here, the requester's connection with Montparnasse can be treated as one of his attributes.

In order to get more information, Alice may enlarge the searching area to the whole city, i.e., Paris in Figure 2. We assume that a user can only be linked to a place's public information, but not to a city's public information. For example, a user's photo can be labeled with any street or square of a city, but not the city itself. The policy can then be written as

$$\bigcirc_{\mathsf{req}} \rhd_{IsLocation} \langle \mathit{is\text{-}in} \rangle \mathsf{Paris}.$$

Here, $\langle \mathit{is\text{-}in} \rangle$ represents a 1-depth relationship path in $\mathcal{PG}$. Depending on the policy, the length of the path can be arbitrary. Note that the requester's connection with Paris can be also formalized as an attribute. However, in this way, each user will be affiliated with a huge number of attributes in the model which may not be an ideal solution.

Scenario 2. In this scenario (the second one in Section 1), Bob wants to use the OSN to organize a fundraising party for children's rare diseases. He intends to let people who are affiliated with at least a certain number, such as three, of different charities as himself to access the event page. The policy is defined as follows.

$$\begin{array}{l}\bigcirc_{\mathsf{own}} \rhd_{IsCharity} \blacktriangledown_{y_1} \blacktriangleright (\mathsf{req} \wedge \\ \bigcirc_{\mathsf{own}} \rhd_{IsCharity} \blacktriangledown_{y_2} (\neg y_1 \wedge \blacktriangleright(\mathsf{req} \wedge \\ \bigcirc_{\mathsf{own}} \rhd_{IsCharity} \blacktriangledown_{y_3} (\neg y_1 \wedge \neg y_2 \wedge \blacktriangleright \mathsf{req}))))\end{array}$$

The left part of Figure 3 depicts an example of three charities ('UNICEF', 'Red Cross' and 'SOS Children's Villages') in $\mathcal{PG}$ needed between a qualified requester and Bob. It can be thought as a public information version of '3-common friends' policy in $\mathcal{UG}$. Three variables, i.e., y_1, y_2 and y_3, mark three charities that Bob is linked with; the conjunction of their negative forms, i.e., $\neg y_1$ and $\neg y_1 \wedge \neg y_2$, in the formula makes sure that these three charities are different.

With our logic, more complicated policies can be achieved based on the information of $\mathcal{PG}$. Suppose that Bob wants to organize another fundraising party for homeless children in Syria during its current civil war. For security and privacy reasons, he believes that the qualified requesters to attend this event should be people who are linked with at least two charities as he is, such as 'UNICEF' and 'Red Cross', that are involved in the humanity aid in Syria organized by the United Nations, i.e., 'Unocha.Syria' in $\mathcal{PG}$,[6]. The policy is defined as

$$\begin{array}{l}\bigcirc_{\mathsf{own}} \rhd \blacktriangledown_{y_1} \langle donate \rangle \blacktriangledown_{y_5} (\mathsf{Unocha.Syria} \wedge \langle donate^{-1} \rangle \blacktriangledown_{y_3} \blacktriangleright (\mathsf{req} \wedge \\ \bigcirc_{\mathsf{own}} \rhd \blacktriangledown_{y_2} (\neg y_1 \wedge \langle donate \rangle (y_5 \wedge \langle donate^{-1} \rangle \blacktriangledown_{y_4} (\neg y_3 \wedge \blacktriangleright \mathsf{req})))))\end{array}$$

The connections between the requester and Bob are shown in the right part of Figure 3. Variables y_1 and y_2 mark two different charities; so do y_3 and y_4 for the requester. We notice that the charities that Bob is related to need not to be different from the ones of the requester. Variable y_5 guarantees that all these organizations have contributions to 'Unocha.Syria'.

Since the public information and their relationships are extracted from external sources, complicated relationship paths in $\mathcal{PG}$ as shown in this example give rise to more meaningful and expressive access control policies.

Scenario 3. In the third scenario in Section1, Charlie only allows his friends who work in the rival company of his employer to view his photo. The policy is

[6] `http://syria.unocha.org/`

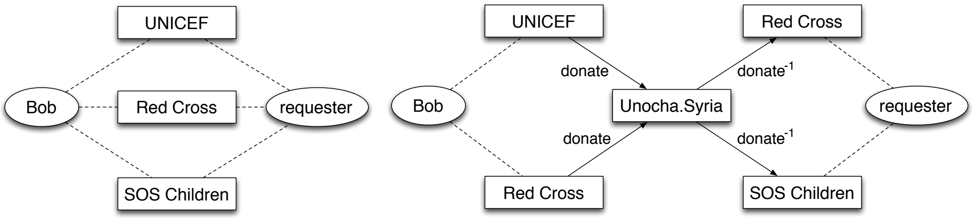


Figure 3: Connections between Bob and qualified requesters.

formally defined as below:

$$\bigcirc_{\mathsf{own}}(\langle \mathit{friend}\rangle \mathsf{req} \wedge (\triangleright\langle \mathit{rival}\rangle \blacktriangleright \mathsf{req})).$$

Different from policies in the previous scenarios, this one requires that the owner and the requester are linked through information in both $\mathcal{UG}$ and $\mathcal{PG}$. More precisely, the sub-formula $\triangleright\langle \mathit{rival}\rangle\blacktriangleright$ regulates that the qualified requester need to work for Company B's rival, i.e., Company A; and the sub-formula $\langle \mathit{friend}\rangle$ filters out the requester who is not a friend of Charlie. We use a conjunction symbol to combine these two parts. In Figure 2, only Alice is qualified as she is a friend of Charlie and she works for Company A.

Scenario 4. In the fourth scenario, suppose that Bob wants to organize another fundraising event, and he wants to invite people who used to participate in the same charities as him and their friends to attend the event. The policy formula is specified as below:

$$\bigcirc_{\mathsf{own}} \triangleright_{IsCharity} \blacktriangleright(\mathsf{req} \vee \langle \mathit{friend}\rangle \mathsf{req}).$$

In Figure 2, Alice is invited to participate this event since she is linked with Bob through a charity (UNICEF). Moreover, Frank, Gabriele and Charlie can also receive the invitation due to their friendships with Alice. Here, the path that links Frank (as well as Gabriele and Charlie) and Bob is composed by both public information and users in the social network model.

7. Using Category Relation in Access Control

In this section, we explore the category relation among public information and incorporate it in our hybrid logic for the aim of concisely specifying access control policies based on public information.

7.1. The Category Relation in Public Information Graph

Let us first consider another scenario. In the model depicted in Figure 2, Charlie is linked with several kinds of sports including Basketball and Tennis. Alice is also a sport fan and her favorite one is Tennis, while Danny likes Volleyball. Charlie has a photo depicting him playing tennis. He only wants his friends who are linked with Tennis to view it. The policy can be defined as

$$\bigcirc_{\mathsf{own}}\langle \mathit{friend}\rangle(\mathsf{req} \wedge (\triangleright\mathsf{Tennis})).$$

Since Alice likes Tennis, she can view the photo. Now, Charlie decides to relax the restriction such that the qualified requester should be his friend who likes any kinds of sports. He modifies his policy as follows:

$$\bigcirc_{\mathsf{own}}\langle \mathit{friend}\rangle(\mathsf{req} \wedge \rhd(\langle \mathit{is}\text{-}a\rangle \mathsf{Sports})).$$

Relationship path $\langle \mathit{is}\text{-}a\rangle$ in the formula marks all the public information that are in an *is-a* relation with Sports in $\mathcal{PG}$, e.g., Tennis. However, this policy cannot achieve Charlie's goal. For example, Danny is not able to view this photo even he is supposed to be. This is because Volleyball is not linked with Sports but Team Sports in *is-a* relationship as shown in Figure 2. In order to grant access to Danny, Charlie again modifies the policy as follows:

$$\bigcirc_{\mathsf{own}}\langle \mathit{friend}\rangle(\mathsf{req} \wedge \rhd(\langle \mathit{is}\text{-}a\rangle \mathsf{Sports} \vee \langle \mathit{is}\text{-}a\rangle\langle \mathit{is}\text{-}a\rangle \mathsf{Sports})).$$

However, there exists many public information related to Sports in the OSN and defining a policy by enumerating all possible lengths is not an acceptable solution. In Wikipedia, articles are organized by means of categories and all the categories form an acyclic graph. Figure 4 shows a part of the category graph of Wikipedia.[7] An article is under (at least) one category, some article can be the main article of a category. For example, article basketball is under the category team sports, it is also the main article of the category basketball. An article under a category is linked with the category's main article. Actually, this is the *is-a* relationship among public information in $\mathcal{PG}$, we call it *category relation*. Since all categories of Wikipedia form an acyclic group (*category graph*), public information together with *is-a* relationships among them compose an acyclic graph as well. For example, the subgraph in the dashed box in Figure 2 is a tree. Next, we integrate the category relation into our logic formula to express above policies in a concise way.

7.2. Logic with the Category Relation

In the model depicted in Figure 2, Charlie is linked with several kinds of sports including Basketball and Tennis. Alice is also a sport fan and her favorite one is Tennis, while Danny likes Volleyball. Charlie has a photo that he wants to share with all his friends who like sports. As depicted in the dash box of Figure 2, these kind of public information are organized by categories. Instead of defining a policy to specify all the sports that are linked to users, we can directly use these category information to define policies.

To make use of the category relations among public information, We first introduce a function on $\mathcal{PG}$ and a new symbol in our logic. The function cf is formally defined as

$$cf(\{f_c\}) = \begin{cases} \{f_c\} & \nexists f_d \text{ s.t. } (f_d, f_c) \in \mathit{is}\text{-}a \\ \bigcup cf(\{f_d\}) & \forall f_d \text{ s.t. } (f_d, f_c) \in \mathit{is}\text{-}a \end{cases}$$

[7] http://en.wikipedia.org/wiki/Help:Categories

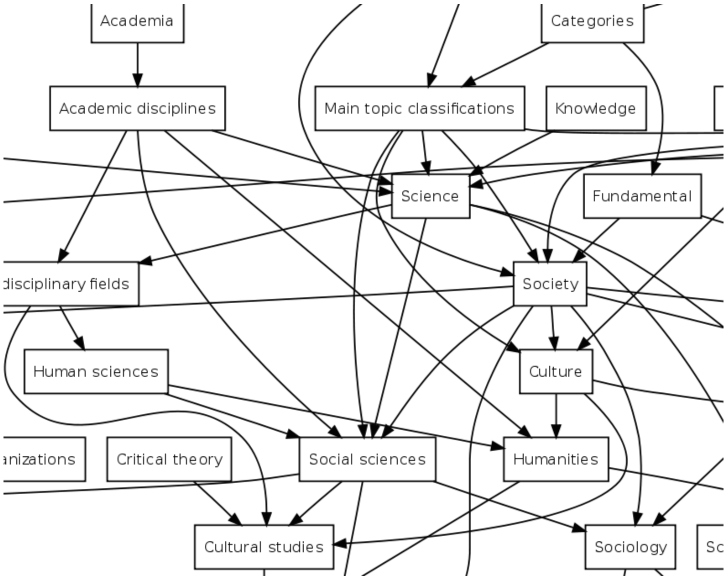


Figure 4: Part of the category hierarchy of Wikipedia.

The result of $cf(\{f_c\})$ contains f_c and all its descendants in an acyclic graph based on *is-a* relationships in $\mathcal{PG}$.

In our hybrid logic, nominal n can represent name of any public information in $\mathcal{PG}$. In order to refer to the node named n as well as all its descendants in the formula, we add a *category nominal* $\lfloor n \rfloor$ into our logic. The syntax of formulas ψ is extended as follows:

$$\psi \ ::= \ t \mid \lfloor n \rfloor \mid q \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \langle j \rangle \psi \mid \bullet_t \psi \mid \blacktriangledown_y \psi \mid \blacktriangleright \phi.$$

The semantics of $\lfloor n \rfloor$ is

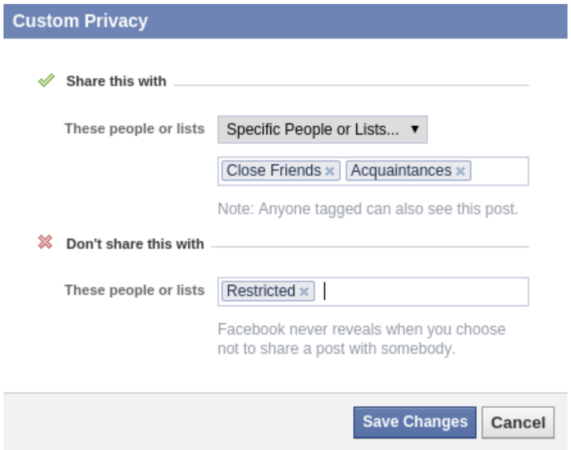
$$\Gamma, \Delta, \rho, \varrho, f_c, \tau \vDash \lfloor n \rfloor \text{ iff } f_c \in cf(V_P(n)) \bigcup V_P(n).$$

With the category nominal, Charlie can easily redefine his policy in the previous example as

$$\bigcirc_{\mathsf{own}} \langle \mathit{friend} \rangle (\mathsf{req} \wedge \triangleright \lfloor \mathsf{Sports} \rfloor).$$

Now, all friends of Charlie who are related to any kind of sport activities, such as Alice and Danny, can access the photo.

Similar to the ones with their contents from Wikipedia, public information from geography databases, i.e., places, together with *is-in* relationships among them also naturally compose an acyclic graph. Therefore, we are able to define policies to qualify the requester, such as “only my friends who have ever been to Europe”, in a concise way without listing different length of *is-in* relationship paths in $\mathcal{PG}$. Other types of hierarchical relationships on public information can also be investigated for the same purpose.



Custom Privacy

✓ Share this with

These people or lists: Specific People or Lists... ▼

Close Friends × Acquaintances ×

Note: Anyone tagged can also see this post.

✗ Don't share this with

These people or lists: Restricted ×

Facebook never reveals when you choose not to share a post with somebody.

Figure 5: Access control with close friends acquaintances and restricted in Facebook.

8. Relationship Hierarchy

In this section, we extend our hybrid logic to capture the hierarchy among different relationships, enabling policy propagation in our access control scheme.

8.1. Relationship Hierarchy

Our social graph model supports multi-relationships. As depicted in Figure 2, Gabriele and Danny are brothers and Alice and Danny are schoolmates. In general, different relationships have different social strength. Family-related relationships, such as spouse and parents, are normally considered stronger than professional relationships such as colleagues. When an owner allows others who are in a certain relationship with him to view one of his resources, those who are in a stronger relationships with the owner intuitively should be able to access the resource as well. For example, if Alice allows her colleagues to view her education background, then her husband and parents should also be able to see it.

In our hybrid logic, to express this kinds of policy, we can define a formula for each relationship type and connect these formulas together with the disjunction operator $\vee$. The policy formula for the above example in our hybrid logic can be specified as

$$\bigcirc_{\mathsf{own}}\langle colleague\rangle\mathsf{req} \vee \bigcirc_{\mathsf{own}}\langle wifeof\rangle\mathsf{req} \vee \bigcirc_{\mathsf{own}}\langle childof\rangle\mathsf{req}.$$

However, this solution is not ideal since it requires the owner to specify the policy for all the intended relationships one by one. It is very likely that the owner misses some relationships, thus the policy cannot fully capture his intention. Therefore, we need a straightforward way to let the owner only specify one relationship in the policy and all the users who are in a stronger relationship with him can access the resource directly. In fact, Facebook already allows a user to put his friends into three (smart) friend lists including “close friend”, “acquaintances” and “restricted” based on their social strength. However, as depicted in Figure 5, a Facebook user still needs to specify these lists in the audience selector (see Section 3) to control who can view his resource, i.e., access control based on social strength is not implemented automatically in Facebook.

To express this kinds of policies in the hybrid logic, we first need to define a hierarchy on all the relationships supported by the OSN. This hierarchy can be built at a system level or a user level. At a system level, OSN operators could regulate the order of relationship types with respect to their social strength. On the other hand, different users may have different opinions about the strength of the relationships. For example, some users believe that college friends are more important than colleagues from work while some have the opposite opinion. Therefore, OSNs could delegate this right to each user and let them freely define the relationship hierarchies themselves.

Here, for the sake of simplicity, we simply assume that the relationship hierarchy is defined at a system level. This indicates that all users in the OSN will share the same relationship hierarchy. The definition of the relationship hierarchy is given as follows.

Definition 1. *A relationship hierarchy is defined as* $(\mathcal{UR}, \leq)$*, where* $\mathcal{UR} = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ *is the relationship type set and* $\leq$ *is a binary relationship on* $\mathcal{UR}$ *which is reflexive, antisymmetric and transitive.*

By its definition, a relationship hierarchy is a partially ordered set. For two relationship types, $\alpha_1 \leq \alpha_2$ indicates that α_2 is a closer relationship than α_1. Figure 6 gives an example of the hierarchy. In this example, spouse is considered the strongest relationship followed by close friends and family. Note that the actual strength of the relationships is out of the scope of this work, OSN operators can follow any theory from the area of sociology to construct the relationship hierarchy.

8.2. Logic with Relationship Hierarchy

To exploit the information in relationship hierarchy to specify our access control policies, we introduce a new symbol $\lceil\langle\alpha_i\rangle\rceil\phi$ into our syntax. The syntax of the user formula is extended to:

$$\begin{aligned} s &::= m \mid x \\ \phi &::= s \mid p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \langle\alpha_i\rangle\phi \mid \lceil\langle\alpha_i\rangle\rceil\phi \mid \bigcirc_s\phi \mid \triangledown_x\phi \mid \rhd\psi. \end{aligned}$$

The semantics of $\lceil\langle\alpha_i\rangle\rceil\phi$ is defined below.

$$\begin{aligned} \mathit{\Gamma}, \mathit{\Delta}, \rho, \varrho, u_a, \tau \vDash \lceil\langle\alpha_i\rangle\rceil\phi \text{ iff } &\exists\, u_b \in \mathcal{U} \text{ s.t. } (u_a, u_b) \in \alpha_j \text{ where } \alpha_i \leq \alpha_j \\ &\wedge\, \mathit{\Gamma}, \mathit{\Delta}, \rho, \varrho, u_b, \tau \vDash \phi \end{aligned}$$

Here, u_b can be in any relationship that is at least the same level of α_i with u_a defined in the relationship hierarchy. To evaluate the policy, the relationship hierarchy should be included in the model $\mathit{\Gamma}$ as well.

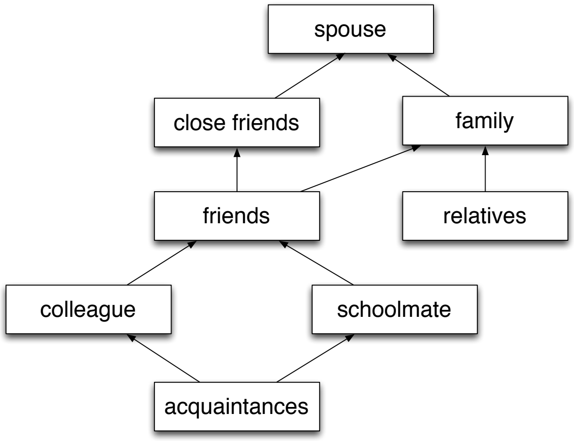


Figure 6: A relationship hierarchy example.

Example 1. Now, with the new operator, an owner could define a policy regulating that users who are at least his colleagues can view one of his resource as

$$\bigcirc_{\mathsf{own}}\lceil\langle colleague\rangle\rceil\mathsf{req}.$$

In addition, the hierarchy operator can be aligned together to express relationship path as well. For example, the following policy means that the requester has to be 3-depth away from the owner and the relationship on each step has to be at least colleague:

$$\bigcirc_{\mathsf{own}}\lceil\langle colleague\rangle\rceil\lceil\langle colleague\rangle\rceil\lceil\langle colleague\rangle\rceil\mathsf{req}.$$

Example 2. To give another example on how to use the hierarchical relationships, recall the social network depicted in Figure 2, suppose that Danny wants to share his interest, such as Volleyball, with his friends. It is clear from Figure 2 that only Charlie can view the information. If Danny intends to share it with users who are also in stronger relationships with him, e.g., Eve (his wife) and Gabriele (his brother), then the policy without using relationship hierarchy will be defined below, where Danny has to explicitly enumerate all the relationships that he considers stronger than freind:

$$\bigcirc_{\mathsf{own}}\langle friend\rangle\mathsf{req} \vee \bigcirc_{\mathsf{own}}\langle husbandof\rangle\mathsf{req} \vee \bigcirc_{\mathsf{own}}\langle brotherof\rangle\mathsf{req}.$$

Now, given the extended logic that supports hierarchical information, Danny could simply redefine the policy in a more concise way:

$$\bigcirc_{\mathsf{own}}\lceil\langle friend\rangle\rceil\mathsf{req}.$$

Moreover, if Danny considers schoolmate a stronger relationship than friend which is different from the hierarchy presented in Figure 6, then Alice can access the resource as well. In this case, instead of using the system level relationship hierarchy, Danny could define his own relationship hierarchy with *friend* $\leq$ *schoolmate* specified. In general, with the extension, our logic can support any hierarchical relationships when defining access control policies.

The main difference between relationship hierarchy and category relationship introduced in Section 7 is the following: relationship hierarchy is defined on relationships, it can only grant access to users who are at the certain distance (specified in the policy) but in different relationships with the owner; on the other hand, category relationship is defined on the nodes in public information graph and it can represent paths of different length in a policy (through the recursively defined function $cf(\{f_c\})$). Further combination of the category relation and relationship hierarchy can be achieved as well, which will give rise to a more powerful way to specify complicated policies in a simple form.

Example 3. To give an example, in Figure 2, suppose that Bob wants to organize a fundraising event, he plans to invite all users who have involved in any fundraising event for Syria before, and their friends (or stronger relationships than friends). This policy exploits both category information related to public information graph and relationship hierarchy related to social graph. By combining the two extensions we have proposed, Bob can simply define the policy as

$$\bigcirc_{\mathsf{own}} \rhd \lfloor \mathsf{Unocha.Syria} \rfloor \blacktriangleright (\mathsf{req} \vee \lceil \mathit{friend} \rceil \mathsf{req}).$$

In Figure 2, Alice and Eve can see the invitation since they are directly involved in some fundraising events (through category relationship). Besides, Frank, Gabirele and Charlie as friends of Alice can join as well. Due to the power of relationship hierarchy, Danny can access the information since he is Eve's husband (following the relationship hierarchy in Figure 6). On the other hand, without the extensions related to category and relationship hierarchy, to define a policy like this, Bob's policy formula will become much longer. We conclude that both extensions improve the concision of our access control scheme.

9. Information Reliability

Owners define policies to control access to their resources. However, in some cases, if the information in OSNs are not reliable, malicious users can still gain access to some resources that they are not supposed to under certain policies. For example, in Scenario 0 of Section 6, If an adversary is able to become friends with three friends of Eve, then he is able to gain the access. Similarly in Scenario 3 of Section 6, a colleague of Charlie, who is also his friend, can maliciously specify that he works for the rival company in the OSN to access Charlie's sensitive photo. As introduced in Section 4, our OSN model contains three parts, i.e., $\mathcal{UG}$, $\mathcal{PG}$ and two maps ρ and ϱ. We discuss about their reliability one by one.

Reliability of $\mathcal{UG}$. Information contained in $\mathcal{UG}$ are mainly users and their relationships. Since a user can describe who he is in the OSN, we only focus on users relationships. To increase user relationships' reliability, we explore trust.

In contrast to the real life, trust between users in OSNs can be quantified, i.e., it has a value. We first add trust values into $\mathcal{UG}$. When u_a establishes an α_i relationship with u_b, u_a will assign a trust value $t_{ab}^{\alpha_i}$ to this relationship. The edge from u_a to u_b is then defined as $(u_a, u_b, \alpha_i, t_{ab}^{\alpha_i})$. Similarly, the edge from u_b to u_a is $(u_b, u_a, \alpha_i^{-1}, t_{ba}^{\alpha_i^{-1}})$. Note that $t_{ab}^{\alpha_i}$ is only known to u_a and $t_{ba}^{\alpha_i^{-1}}$ is only known to u_b, and these two values can be different. We regulate that every trust value is in the interval $[0, 1]$, the bigger the value is, more trust it represents. We additionally introduce two new operators $\langle\alpha_i\rangle^{\rightarrow t}\phi$ and $\langle\alpha_i\rangle^{\leftarrow t}\phi$ into the user formula ϕ and their semantics are defined as follows.

$$\begin{array}{ll}
\mathit{\Gamma}, \mathit{\Delta}, \rho, \varrho, u_a, \tau \vDash \langle\alpha_i\rangle^{\rightarrow t}\phi \text{ iff} & \exists\, u_b \in \mathcal{U} \text{ s.t. } (u_a, u_b) \in \alpha_i, t_{ab}^{\alpha_i} \geq t \text{ and} \\
& \mathit{\Gamma}, \mathit{\Delta}, \rho, \varrho, u_b, \tau \vDash \phi \\
\mathit{\Gamma}, \mathit{\Delta}, \rho, \varrho, u_a, \tau \vDash \langle\alpha_i\rangle^{\leftarrow t}\phi \text{ iff} & \exists\, u_b \in \mathcal{U} \text{ s.t. } (u_b, u_a) \in \alpha_i^{-1}, t_{ba}^{\alpha_i^{-1}} \geq t \text{ and} \\
& \mathit{\Gamma}, \mathit{\Delta}, \rho, \varrho, u_b, \tau \vDash \phi
\end{array}$$

When the requester is regulated to be linked with the owner through user relationships, trust can be put into the formula. Now for the policy of Scenario 0, Eve can specify the formula as below:

$$\bigcirc_{\mathsf{own}}\langle \mathit{friend}\rangle_3^{\rightarrow 0.8}\mathsf{req}.$$

To get an illegal access with the above formula, a malicious user needs to become friends with three users that Eve trusts ($t \geq 0.8$). Note that the way we integrate trust value into the user formula is simple. There exist other methods, such as trust value can be evaluated on a whole relationship path. How to extend our logic to support complicated trust requirements is part of our future work.

Reliability of $\mathcal{PG}$. Different from users' information, public information are imported from external databases and they are not operated by real users. For example, Paris's information in Facebook is taken from Wikipedia and the fact that it is in France can be extracted from public geography database. Therefore, reliability of public information are guaranteed by these external sources – for instance, the reliability of Wikipedia pages and their connections can be ensured by a community effort and users' reputation [41].

Reliability of ρ and ϱ. Some public information result in user relationships, for example, users who went to the same school are 'schoolmates' or work in the same company are 'colleagues'. If the link between the qualified requester and this kind of public information are exploited by a policy, then the owner who defines this policy can add the connection originated by the public information between the qualified requester and other users into the formula as well. In this way, these other users can be treated as endorsing the connection between the requester and the public information. In Scenario 3 of Section 6, besides working in the rival company, Charlie regulates that the qualified requester should have a certain number, e.g., 3, of colleagues who work in this rival company. Moreover, he can also add trust to the formula. The policy is defined as follows.

$$\bigcirc_{\mathsf{own}}(\langle \mathit{friend}\rangle^{\rightarrow 0.8}\mathsf{req} \wedge (\triangleright\langle \mathit{rival}\rangle\blacktriangledown_y \blacktriangleright (\mathsf{req} \wedge \langle \mathit{colleague}\rangle_3^{\leftarrow 0.7} \triangleright y))).$$

(co-)owner	Policy formula ϕ	Qualified users
Alice	$\bigcirc_{\mathsf{own}}\langle friend\rangle\mathsf{req}$	Frank, Charlie
Bob	$\bigcirc_{\mathsf{own}}\langle friend\rangle\mathsf{req}$	Eve
Gabriele	$\bigcirc_{\mathsf{own}}\lceil\langle friend\rangle\rceil\mathsf{req}$	Eve, Danny

Table 1: (Co-)owners with their policies and users who can access the resource.

Now, in order to gain the access, the malicious user has to be trusted by Charlie ($t \geq 0.8$) and be colleagues with three other users who work in that company. Also, these three colleagues' trust value on the requester have to be at least 0.7. Clearly, it is much harder for the adversary to succeed.

For policies exploiting public information that cannot result in user relationships, endorsement (as well as trust) cannot be applied. For example, in Scenario 1 of Section 6, the qualified requester needs to be linked to a location, while in Scenario 2 Bob and the requester are connected through charities. Similar to public information, the reliability of the links between some of these public information and users also depends on external services. For example, in Facebook, a user is treated as having been to one location if he used to publish a status or photo labeled with that location. This location label is provided by ISP (Internet Service Provider) or GPS services. A user's connection to a charity can be certified by the charity, as the user normally gets tax benefit for his donations. Again, we do not focus on the reliability of external services.

10. Collaborative Access Control

So far, we have assumed that the resource's access control policy can be only defined by its owner. However, as introduced in Section 5, a resource can be affiliated with several users, e.g., a photo tagged with several users, and each of them should have the right to decide who can access the resource. This is the so-called collaborative access control. In this section, we aim to extend our model to support collaborative access control.

We first name all the users who are affiliated with a resource and are not the owner as the *co-owners* of the resource. We further use the set $O(r)$ to represent a resource r's owner and co-owners. If one co-owner of a resource wants to define a policy to allow only his friends of friends to view the resource, then the policy formula is specified as $\bigcirc_{\mathsf{own}}\langle friend\rangle\langle friend\rangle\mathsf{req}$. For simplicity, we still use variable own in the formula to refer to one of the co-owners in $O(r)$.

With multiple policies on a resource, access control conflicts can happen when deciding whether granting the access to a certain user or not. Informally, a conflict means a user can access the resource under one policy but is forbidden by another. For example, in the user graph depicted in Figure 2, suppose that Alice publishes a photo and tags her friends Bob and Gabriele in it. Here, Alice is the owner while Bob and Gabriele are the co-owners of the photo. We assume that Alice and Bob only allow their friends to view this photo and Gabriele wants users who are at least his friends to view it (see Section 8). Their policy formulas as well as users who can access the photo, namely qualified users, are listed in Table 1. There are several access control conflicts. For example, Eve can access the resource under Bob and Gabriele's policies but she is forbidden by Alice. Note that the owner and co-owners of resource can always access the resource, and they are not included in the qualified users of each policy.

To formalize access control conflicts, we first define the set of qualified users of a policy as the following.

Definition 2. *Given an access control policy ϕ that is defined by a user u on a resource r, i.e., $u \in O(r)$, its set of qualified requesters is $\mathcal{QU}(\phi) = \{u' \mid \mathit{\Gamma}, \mathit{\Delta}, \rho, \varrho, u, \tau[\mathsf{own} \mapsto u, \mathsf{req} \mapsto u'] \vDash \phi \wedge u' \notin O(r)\}$.*

Then, the *conflict* on accessing a resource is defined as

Definition 3. *Given a resource with the set of access control policies defined on it, denoted by $\mathit{\Phi}$. An access control conflict happens if there exists $u \in \mathcal{QU}(\phi)$ for a policy $\phi \in \mathit{\Phi}$ such that $u \notin \mathcal{QU}(\phi')$ for another policy $\phi' \in \mathit{\Phi}$.*

Several works have been proposed to resolve conflicts caused by collaborative access control (see Section 2 for a short introduction), we can apply some of them within our scheme. For instance, Hu et al. [35] proposed a few solutions for resolving access control conflicts. In their work, the so-called *naive* solution is to only allow the common users in the sets of qualified requesters to access the resource. In the example of Table 1, no one except for the co-owners can view the photo. This shows that the *naive* solution is too restrictive. In addition, more sophisticated solutions based on voting schemes are proposed by Hu et al. [35] and others [32].

So far, we have considered conflicts at the requester level, i.e., conflicts happen when different co-owners allow different users to access the resource. In [34], Sun et al. considered conflicts at a policy level and proposed an approach for resolving conflicts by combining trust relations in OSNs and preferential voting schemes. Under their consideration, a conflict happens when co-owners' policies are different. In Figure 2, following the example in this section, Alice, Bob and Gabriele co-own a photo. Since the policies listed in Table 1 from them are different, a policy-level conflict happens. The solutions to resolve the requester-level conflicts can be naturally exploited to resolve the policy-level ones. For instance, one naive solution would be: only the owner's policy is enforced on controlling the photo's access. In this case, Gabirele's policy is ignored.

We notice that, in some cases, there are no policy-level conflicts but requester-level ones. For example, in Table 1, Alice and Bob have the identical policy, thus there is no policy-level conflict between them. On the other hand, as we discussed before, their policies still cause requester-level conflicts. In some other cases, there may be no requester-level conflicts but policy-level ones. For instance, suppose that in Figure 2 Alice and Charlie are tagged in a same photo when they watched a Tennis game at school several years ago. Charlie wants to share this photo with his friends who like sports, i.e., $\bigcirc_{\mathsf{own}}\langle \mathit{friend} \rangle(\mathsf{req} \wedge \rhd \lfloor \mathsf{Sports} \rfloor)$. In Figure 2, except for Alice, only Danny is

	[10]	[13]	[21]	This paper
Multi-relationship type	✓	✓	✓	✓
User attributes	✓	✓		✓
Public information				✓
Trust		✓		✓
User-resource relation			✓	
Relationship depth	✓	✓	✓	✓
Topology-based policy	✓			✓
Policy propagation		✓		✓

Table 2: Comparison of access control schemes for OSNs.

qualified. Alice, however, only wants to share this photo with her schoolmates, In Figure 2, only Danny can view the photo under Alice's policy. There is no conflict at the requester-level since the only qualified requester is Danny. However, Alice and Charlie's policies are obviously different which results in a policy-level conflict. The relationship between these two types of conflicts deserves further investigations, we leave it as a future work.

11. Comparison

In this section, we compare our scheme with relationship-based access control schemes in the literature [10, 13, 21] (see Table 2).

The model of OSNs in [10] is the same as our user graph $\mathcal{UG}$, but public information are not treated as entities in the model. As a consequence, access control policies only make use of users' social representations. On the other hand, it seems possible to express connections between users and public information through propositions in [10]. For example, a proposition *IsinParis* can be used to express the connection between a user and city Paris. However, as mentioned in Section 6, each user will be affiliated with a large amount of attributes which is neither ideal or practical. Moreover, policies that explore relationships between public information (see examples in Section 6), cannot be captured by propositions.

The work proposed in [13] does not explicitly take into account public information and their relationships. However, this work has two interesting features. First, in the OSN model, users' resources are treated as independent entities. Relationships between users and resources are not restricted only to ownership, e.g., the relationship between a user and a photo that he is tagged in is modeled as 'photoOf' in their language. Thus, collaborative access control is possible in their model. Second, due to the fact that OSNs are modeled with semantic web technologies, hierarchy information among users' relationships are naturally supported as well as actions and resources, which make policy propagation possible. For example, if a user defines a policy to regulate the qualified requester to be his friends, then users who are in a closer relationship, such as 'good friend', with him are also qualified. In our work, we show how to perform policy propagation based on a model of relationship hierarchy in our access control scheme (see in Section 8). In addition, we used semantic relations among the public information in Section 7 to facilitate users to express their policies concisely.

Similarly, the scheme in [21] does not take into account public information neither. In this model, attributes of users are not represented. Moreover, their policy language seems weaker than ours – negation symbol only works with relationship paths, but not on nodes. Hence, policies such as "all my friends but Alice can view my photo" cannot be expressed. On the other hand, this work has some its own features. First, the OSN model treats resources as nodes which is similar to the one in [13], and actions that users performed on their resources are recognized as relationships. For example, a user can regulate that only users who used to comment on a same photo as he did is able to poke him. To support this in our access control model, we need to extend the social network model and treat users' resources as nodes as well. Second, the authors propose a simple solution through administrative policies for collaborative access control. To achieve this in our model, we need to add a decision module in the model checking algorithm.

We also notice that the two schemes [13, 21] can possibly treat public information as users' resources, i.e., modeled as nodes in their OSN model. However, as we explained previously in Section 4, public information are extracted often from external databases, and relationships among them are different from the ones between users. In our work, we apply the *separation of concerns* principle to model public information and their relationships separately from users and their social links.

12. Discussion

We have shown that our scheme and its extensions can express fine-grained access control policies related to users and public information. We have also shown how to deal with the problem of information reliability in OSNs by incorporating endorsement and trust into our policy formulas. There are still two other issues to discuss.

The first question is about the *usability* of our scheme, especially for the non-experienced users – whether a user can easily express a policy of his intention. On one hand, relationship-based policies (e.g., friends, friends of friends) can be easily expressed in our scheme like the current access control schemes adopted by OSNs. On the other hand, a group of qualified requesters under a sophisticated policy can be computed by OSNs, e.g., a Facebook user can directly get a list of his friends who have been worked in a company through Graph Search. Besides, as shown in Figure 1, Facebook already implemented smart list for users to define fine-grained policies. Therefore, we believe that our scheme can be supported as well. Moreover, users can use visualization tools (e.g., see [42]) to learn whether their policies have been properly enforced.

The second is related to the *availability* of user information in OSNs. As privacy raises serious concerns in OSNs, users might not be willing to share too much information. As a consequence, some eligible users can be filtered out by a policy due to the lack of their information in the OSN. However, the main purpose of OSNs is for people to express themselves and socialize with other users – more information a user shares, more benefits he will gain from the OSN. On the contrary, a user keeps more privacy if he shares less information. There is always a balance between information sharing (or utility) and privacy. What we focus in this paper is to explore the information shared by users in OSNs to express fine-grained access control policies. Thus, we consider availability of user information in OSNs orthogonal to our proposal.

13. Conclusion and Future Work

In this paper, we have first identified a new type of access control policies that are meaningful but have never been addressed in the literature. Namely, users in OSNs can express access control requirements not only based on their social relations but also on their connections through public information. Then we defined an OSN model containing users and public information, based on which we proposed a hybrid logic to define access control policies. We gave a number of policies based on public information and formulated them formally and precisely in our proposed logic. We further used category relations among public information and relationship hierarchy to extend our logic and make it more practical. In addition, we also showed how to extend our model and logic to deal with unreliable information and collaborative access control in OSNs.

In the future, besides studying the relationship between requester-level and policy-level conflicts related to collaborative access control framework (see Section 10), we plan to improve the expressiveness of our model by integrating user resources [13, 21]. As resources are different from users, modeling resources explicitly may address more expressive policies. The connections between resources and public information will be interesting to study as well. Secondly, we plan to develop a Facebook App to support our access control models. This App should guide users to use public information within Facebook to express their intentions on control their resources' access. The main feature of this App is to give users a way to organize their friends into different lists or groups by exploring different public information. Besides, a visualization tool, similar to the one of Anwar and Fong [20], will be developed as part of the App to help users to find and evaluate who else in Facebook can access his resources under his policies.

References

[1] J. Pang, Y. Zhang, A new access control scheme for Facebook-style social networks, in: Proc. 9th Conference on Availability, Reliability and Security (ARES), IEEE CS, 2014, pp. 1–10.

[2] R. S. Sandhu, Lattice-based access control models, IEEE Computer 26 (11) (1993) 9–19.

[3] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman, Role-based access control models, IEEE Computer 29 (2) (1996) 38–47.

[4] M. Abadi, Logic in access control, in: Proc. 18th Annual IEEE Symposium on Logic in Computer Science (LICS), IEEE CS, 2003, pp. 228–233.

[5] M. Abadi, C. Fournet, Access control based on execution history, in: Proc. 10th Annual Network & Distributed System Security Symposium (NDSS), Internet Society, 2003, pp. 107–121.

[6] N. Li, J. C. Mitchell, W. H. Winsborough, Beyond proof-of-compliance: security analysis in trust management, Journal of the ACM 52 (3) (2005) 474–514.

[7] J.-W. Byun, E. Bertino, N. Li, Purpose based access control of complex data for privacy protection, in: Proc. 10th ACM Symposium on Access Control Models and Technologies (SACMAT), ACM, 2005, pp. 102–110.

[8] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, S. Capkun, Proximity-based access control for implantable medical devices, in: Proc. 16th ACM Conference on Computer and Communications Security (CCS), ACM, 2009, pp. 410–419.

[9] D. Liu, N. Li, X. Wang, L. J. Camp, Beyond risk-based access control: towards incentive-based access control, in: Proc. 15th Conference on Financial Cryptography and Data Security (FC), Vol. 7035 of LNCS, Springer, 2012, pp. 102–112.

[10] G. Bruns, P. W. L. Fong, I. Siahaan, M. Huth, Relationship-based access control: its expression and enforcement through hybrid logic, in: Proc. 2nd ACM Conference on Data and Application Security and Privacy (CODASPY), ACM, 2012, pp. 117–124.

[11] C. E. Gates, Access control requirements for Web 2.0 security and privacy, in: Proc. IEEE Workshop on Web 2.0 Security and Privacy (W2SP), 2007.

[12] B. Carminati, E. Ferrari, A. Perego, Enforcing access control in web-based social networks, ACM Transactions on Information & System Security 13 (1) (2009) Article No. 6.

[13] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, B. Thuraisingham, A semantic web based framework for social network access control, in: Proc. 14th ACM Symposium on Access Control Models and Technologies (SACMAT), ACM, 2009, pp. 177–186.

[14] P. W. L. Fong, M. M. Anwar, Z. Zhao, A privacy preservation model for Facebook-style social network systems, in: Proc. 14th European Symposium on Research in Computer Security (ESORICS), Vol. 5789 of LNCS, Springer, 2009, pp. 303–320.

[15] P. W. L. Fong, Relationship-based access control: protection model and policy language, in: Proc. 1st ACM Conference on Data and Application Security and Privacy (CODASPY), ACM, 2011, pp. 191–202.

[16] P. W. L. Fong, I. Siahaan, Relationship-based access control policies and their policy languages, in: Proc. 16th ACM Symposium on Access Control Models and Technologies (SACMAT), ACM, 2011, pp. 51–60.

[17] E. Tarameshloo, P. W. L. Fong, Access control models for geo-social computing systems, in: Proc. 19th ACM Symposium on Access Control Models and Technologies (SACMAT), ACM, 2014, pp. 115–126.

[18] E. Tarameshloo, P. W. L. Fong, P. Mohassel, On protection in federated social computing systems, in: Proc. 4th ACM Conference on Data and Application Security and Privacy (CODASPY), ACM, 2014, pp. 75–86.

[19] M. Cramer, J. Pang, Y. Zhang, A logical approach to restricting access in online social networks, in: Proc. 20th ACM Symposium on Access Control Models and Technologies (SACMAT), ACM Press, 2015.

[20] M. M. Anwar, P. W. L. Fong, A visualization tool for evaluating access control policies in Facebook-style social network systems, in: Proc. 27th ACM Symposium on Applied Computing (SAC), ACM, 2012, pp. 1443–1450.

[21] Y. Cheng, J. Park, R. S. Sandhu, Relationship-based access control for online social networks: beyond user-to-user relationships, in: Proc. 4th IEEE Conference on Information Privacy, Security, Risk and Trust (PASSAT), IEEE CS, 2012, pp. 646–655.

[22] J. Crampton, J. Sellwood, Path conditions and principal matching: a new approach to access control, in: Proc. 19th ACM Symposium on Access Control Models and Technologies (SACMAT), ACM, 2014, pp. 187–198.

[23] B. Carminati, E. Ferrari, Privacy-aware collaborative access control in web-based social networks, in: Proc. 22nd IFIP WG 11.3 Working Conference on Data and Applications Security (DBSEC), Vol. 5094 of LNCS, Springer, 2008, pp. 81–96.

[24] J. Domingo-Ferrer, A. Viejo, F. Sebé, Úrsula González-Nicolás, Privacy homomorphisms for social networks with private relationships, Computer Networks 52 (15) (2008) 3007–3016.

[25] B. Carminati, E. Ferrari, Enforcing relationships privacy through collaborative access control in web-based social networks, in: Proc. 5th Conference on Collaborative Computing (CollaborateCom), IEEE CS, 2009, pp. 1–8.

[26] K. B. Frikken, P. Srinivas, Key allocation schemes for private social networks, in: Proc. 8th ACM Workshop on Privacy in the Electronic Society (WPES), ACM, 2009, pp. 11–20.

[27] G. Mezzour, A. Perrig, V. Gligor, P. Papadimitratos, Privacy-preserving relationship path discovery in social networks, in: Proc. 8th Conference on Cryptology and Network Security (CANS), Vol. 5888 of LNCS, Springer, 2009, pp. 189–208.

[28] M. Xue, B. Carminati, E. Ferrari, P3D - privacy-preserving path discovery in decentralized online social networks, in: Proc. 35th IEEE Computer Software and Applications Conference (COMPSAC), IEEE CS, 2011, pp. 48–57.

[29] M. Backes, M. Maffei, K. Pecina, A security API for distributed social networks, in: Proc. 18th Annual Network & Distributed System Security Symposium (NDSS), Internet Society, 2011, pp. 35–51.

[30] M. Nagy, E. D. Cristofaro, A. Dmitrienko, N. Asokan, A. Sadeghi, Do I know you?: efficient and privacy-preserving common friend-finder protocols and applications, in: Proc. 29th Annual Computer Security Applications Conference (ACSAC), ACM, 2013, pp. 159–168.

[31] J. Pang, Y. Zhang, Cryptographic protocols for enforcing relationship-based access control policies, in: Proc. 39th Annual IEEE Computers, Software & Applications Conference (COMPSAC), IEEE CS, 2015.

[32] A. C. Squicciarini, M. Shehab, F. Paci, Collective privacy management in social networks, in: Proc. 18th Conference on World Wide Web (WWW), ACM, 2009, pp. 521–530.

[33] Y. Sun, C. Zhang, J. Pang, B. Alcalde, S. Mauw, A trust-augmented voting scheme for collaborative privacy management, in: Proc. 6th Workshop on Security and Trust Management (STM), Vol. 6710 of LNCS, Springer, 2011, pp. 132–146.

[34] Y. Sun, C. Zhang, J. Pang, B. Alcalde, S. Mauw, A trust-augmented voting scheme for collaborative privacy management, Journal of Computer Security 20 (4) (2012) 437–459.

[35] H. Hu, G.-J. Ahn, J. Jorgensen, Multiparty access control for online social networks: Model and mechanisms, IEEE Transactions on Knowledge and Data Engineering 26 (7) (2013) 1614–1627.

[36] A. Besmer, H. Lipford, Moving beyond untagging: photo privacy in a tagged world, in: Proc. 28th ACM Conference on Human Factors in Computing Systems (CHI), ACM, 2010, pp. 1563–1572.

[37] P. Klemperer, Y. Liang, M. Mazurek, M. Sleeper, B. Ur, L. Bauer, L. Cranor, N. Gupta, M. Reiter, Moving beyond untagging: photo privacy in a tagged world, in: Proc. 30th ACM Conference on Human Factors in Computing Systems (CHI), ACM, 2012, pp. 377–386.

[38] A. Felt, D. Evans, Privacy protection for social networking platforms, in: Workshop on Web 2.0 Security and Privacy, 2008.

[39] K. Singh, S. Bhola, W. Lee, xBook: Redesigning privacy control in social networking platforms, in: Proc. 18th USENIX Security Symposium, USENIX Association, 2009, pp. 249–266.

[40] M. Shehab, A. Squicciarini, G.-J. Ahn, I. Kokkinou, Access control for online social networks third party applications, Computers & Security 31 (8) (2012) 897–911.

[41] B. T. Adler, L. de Alfaro, A content-driven reputation system for the wikipedia, in: Proc. 16th Conference on World Wide Web (WWW), ACM, 2007, pp. 261–270.

[42] M. M. Anwar, P. W. L. Fong, X.-D. Yang, H. J. Hamilton, Visualizing privacy implications of access control policies in social network systems, in: Proc. 4th Workshop on Data Privacy Management (DPM), Vol. 5939 of LNCS, Springer, 2009, pp. 106–120.