Multiparty Authorization Framework for Data Sharing in Online Social Networks [*]

Hongxin Hu and Gail-Joon Ahn

Arizona State University, Tempe, AZ 85287, USA
{hxhu,gahn}@asu.edu

Abstract. Online social networks (OSNs) have experienced tremendous growth in recent years and become a *de facto* portal for hundreds of millions of Internet users. These OSNs offer attractive means for digital social interactions and information sharing, but also raise a number of security and privacy issues. While OSNs allow users to restrict access to shared data, they currently do not provide effective mechanisms to enforce privacy concerns over data associated with multiple users. In this paper, we propose a multiparty authorization framework that enables collaborative management of shared data in OSNs. An access control model is formulated to capture the essence of multiparty authorization requirements. We also demonstrate the applicability of our approach by implementing a proof-of-concept prototype hosted in Facebook.

Keywords: Social network, Multiparty, Access control, Privacy, Data sharing

1 Introduction

In recent years, we have seen unprecedented growth in the application of OSNs. For example, Facebook, one of representative social network sites, claims that it has over 500 million active users and over 30 billion pieces of shared contents each month [2], including web links, news stories, blog posts, notes and photo albums. To protect user data, access control has become a central feature of OSNs [1, 3].

A typical OSN provides each user with a virtual space containing profile information, a list of the user's friends, and web pages, such as *wall* in Facebook, where users and friends can post contents and leave messages. A user profile usually includes information with respect to the user's birthday, gender, interests, education and work history, and contact information. In addition, users can not only upload a content into their own or others' spaces but also *tag* other users who appear in the content. Each tag is an explicit reference that links to a user's space. For the protection of user data, current OSNs indirectly require users to be system and policy administrators for regulating their data, where users can restrict data sharing to a specific set of trusted users. OSNs often use *user relationship* and *group membership* to distinguish between trusted and untrusted users. For example, in Facebook, users can allow *friends*, *friends of friends*, *specific groups* or *everyone* to access their data, relying on their personal authorization and privacy requirements.

[*] This work was partially supported by the grants from National Science Foundation (NSF-IIS-0900970 and NSF-CNS-0831360) and Department of Energy (DE-SC0004308).

Although OSNs currently provide simple access control mechanisms allowing users to govern access to information contained in their own spaces, users, unfortunately, have no control over data residing outside their spaces. For instance, if a user posts a comment in a friend's space, s/he cannot specify which users can view the comment. In another case, when a user uploads a photo and tags friends who appear in the photo, the tagged friends cannot restrict who can see this photo, even though the tagged friends may have different privacy concerns about the photo. To address such an issue, preliminary protection mechanisms have been offered by existing OSNs. For example, Facebook allows tagged users to remove the tags linked to their profiles. However, removing a tag from a photo can only prevent other members from seeing a user's profile by means of the association link, but the user's image is still contained in the photo. Since original access control policies cannot be changed, the user's image continues to be accessed by all authorized users. Hence, it is essential to develop an effective and flexible access control mechanism for OSNs, accommodating the special authorization requirements coming from multiple associated users for collaboratively managing the shared data.

In this paper, we propose a multiparty authorization framework (MAF) to model and realize multiparty access control in OSNs. We begin by examining how the lack of multiparty access control for data sharing in OSNs can undermine the protection of user data. A multiparty authorization model is then formulated to capture the core features of multiparty authorization requirements which have not been accommodated so far by existing access control systems and models for OSNs (e.g., [6, 7, 13, 14, 19]). Meanwhile, as conflicts are inevitable in multiparty authorization specification and enforcement, systematic conflict resolution mechanism is also addressed to cope with authorization and privacy conflicts in our framework.

The rest of the paper is organized as follows. Section 2 gives a brief overview of related work. In Section 3, we present multiparty authorization requirements and articulate our proposed multiparty authorization model, including multiparty authorization specification and multiparty policy evaluation. Implementation details and experimental results are described in Section 4. Section 5 concludes this paper.

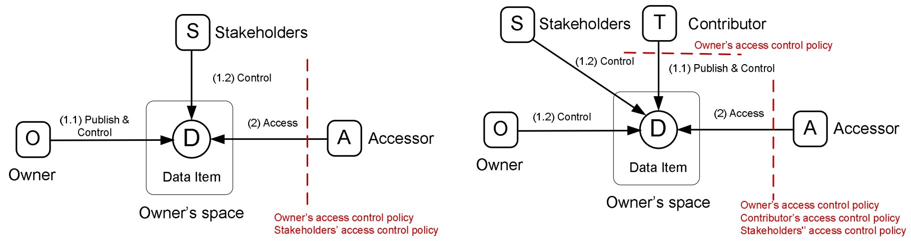
2 Related Work

Access control for OSNs is still a relatively new research area. Several access control models for OSNs have been introduced (e.g., [6, 7, 13, 14, 19]). Early access control solutions for OSNs introduced trust-based access control inspired by the developments of trust and reputation computation in OSNs. The D-FOAF system [19] is primarily a Friend of a Friend (FOAF) ontology-based distributed identity management system for OSNs, where relationships are associated with a trust level, which indicates the level of friendship between the users participating in a given relationship. Carminati et al. [6] introduced a conceptually-similar but more comprehensive trust-based access control model. This model allows the specification of access rules for online resources, where authorized users are denoted in terms of the relationship type, depth, and trust level between users in OSNs. They further presented a semi-decentralized discretionary access control model and a related enforcement mechanism for controlled sharing of information in OSNs [7]. Fong et al. [14] proposed an access control model that formalizes and generalizes the access control mechanism implemented in Facebook, admitting arbitrary policy vocabularies that are based on theoretical graph properties. Gates [8] described relationship-based access control as one of new security paradigms that addresses unique requirements of Web 2.0. Then, Fong [13] recently formulated this paradigm called a Relationship-Based Access Control (ReBAC) model that bases authorization decisions on the relationships between the resource owner and the resource accessor in an OSN. However, none of these existing work could model and analyze access control requirements with respect to collaborative authorization management of shared data in OSNs.

Recently, semantic web technologies have been used to model and express fine-grained access control policies for OSNs (e.g., [5, 10, 21]). Especially, Carminati et al. [5] proposed a semantic web based access control framework for social networks. Three types of policies are defined in their framework, including authorization policy, filtering policy and admin policy, which are modeled with the Web Ontology Language (OWL) and the Semantic Web Rule Language (SWRL). Access control policies regulate how resources can be accessed by the participants; filtering policies specify how resources have to be filtered out when a user fetches an OSN page; and admin policies can determine who is authorized to specify policies. Although they claimed that flexible admin policies are needed to bring a system to a scenario where several access control policies specified by distinct users can be applied to the same resource, the lack of formal descriptions and concrete implementation of the proposed approach leaves behind the ambiguities of their solution.

The need of joint management for data sharing, especially photo sharing, in OSNs has been recognized by the recent work [4, 24, 26]. The closest work to this paper is probably the solution provided by Squicciarini et al. [24] for collective privacy management in OSNs. Their work considered access control policies of a content that is co-owned by multiple users in an OSN, such that each co-owner may separately specify her/his own privacy preference for the shared content. The Clarke-Tax mechanism was adopted to enable the collective enforcement of policies for shared contents. Game theory was applied to evaluate the scheme. However, a general drawback of their solution is the usability issue, as it could be very hard for ordinary OSN users to comprehend the Clarke-Tax mechanism and specify appropriate bid values for auctions. In addition, the auction process adopted in their approach indicates that only the winning bids could determine who can access the data, instead of accommodating all stakeholders' privacy preferences. In contrast, our work proposes a formal model to address the multiparty access control issue in OSNs, along with a general policy specification scheme and a simple but flexible conflict resolution mechanism for collaborative management of shared data in OSNs.

Other related work include general conflict resolution mechanisms for access control [12, 15–18, 20] and learn-based generation of privacy policies for OSNs [11, 22, 23]. All of those related work are orthogonal to our work.



(a) A shared data item has multiple stakeholders

(b) A shared data item is published by a contributor

Fig. 1. Scenarios of Multiparty Authorization in OSNs.

3 Multiparty Authorization for OSNs

In this section, we analyze the requirements of multiparty authorization (Section 3.1) and address the modeling approach we utilize to represent OSNs (Section 3.2). We also introduce a policy scheme (Section 3.3) and an authorization evaluation mechanism (Section 3.4) for the specification and enforcement of multiparty access control policies in OSNs.

3.1 Requirements

OSNs provide built-in mechanisms enabling users to communicate and share data with other members. OSN users can post statuses and notes, upload photos and videos in their own spaces, and tag others to their contents and share the contents with their friends. On the other hand, users can also post contents in their friends' spaces. The shared contents may be connected with multiple users. Consider an example where a photograph contains three users, Alice, Bob and Carol. If Alice uploads it to her own space and tags both Bob and Carol in the photo, we call Alice an *owner* of the photo, and Bob and Carol *stakeholders* of the photo. All of these users may specify access control policies over this photo. Figure 1(a) depicts a data sharing scenario where the owner of a data item shares the data item with other OSN members, and the data item has multiple stakeholders who may also want to involve in the control of data sharing. In another case, when Alice posts a note stating *"I will attend a party on Friday night with @Carol"* to Bob's space, we call Alice a *contributor* of the note and she may want to make the control over her notes. In addition, since Carol is explicitly identified by *@-mention* (at-mention) in this note, she is considered as a *stakeholder* of the note and may also want to control the exposure of this note. Figure 1(b) shows another data sharing scenario where a contributor publishes a data item to someone else's space and the data item may also have multiple stakeholders (e.g., tagged users). All associated users should be allowed to define access control policies for the shared data item.

OSNs also enable users to share others' data. For example, when Alice views a photo in Bob's space and selects to share this photo with her friends, the photo will be in turn posted to her space and she can specify access control policies to authorize her friends to see this photo. In this case, Alice is a *disseminator* of the photo. Since Alice may adopt a weaker control saying the photo is visible to everyone, the initial access control requirements of this photo should be complied with, preventing from the possible leakage of sensitive information via the procedure of data dissemination. For a more complicated case, the disseminated data may be further *re-disseminated* by disseminator's friends, where effective access control mechanisms should be applied in each procedure to regulate *sharing* behaviors. Especially, regardless of how many steps the data item has been re-disseminated, the original access control policies should be always enforced to protect the data dissemination.

3.2 Modeling Social Networks

An OSN can be represented by a relationship network, a set of user groups and a collection of user data. The relationship network of an OSN is a directed labeled graph, where each node denotes a user, and each edge represents a relationship between users. The label associated with each edge indicates the type of the relationship. Edge direction denotes that the initial node of an edge establishes the relationship and the terminal node of the edge accepts the relationship. The number and type of supported relationships rely on the specific OSNs and its purposes. Besides, OSNs include an important feature that allows users to be organized in groups [28, 27], where each group has a unique name. This feature enables users of an OSN to easily find other users with whom they might share specific interests (e.g., same hobbies), demographic groups (e.g., studying at the same schools), political orientation, and so on. Users can join in groups without any approval from other group members. Furthermore, OSNs provide each member with a web space where users can store and manage their personal data including profile information, friend list and user content.

We now formally model and define an online social network as follows:

Definition 1 *(**Online Social Network**). An online social network is modeled as a 9-tuple* $OSN =< U, G, PC, RT, RC, TT, CC, UU, UG >$*, where*

- U *is a set of users of the OSN. Each user has a unique identifier;*
- G *is a set of groups to which the users can belong. Each group also has a unique identifier;*
- PC *is a collection of user profile sets,* $\{p_1, \dots, p_n\}$*, where* $p_i = \{pi_1, \dots, pi_m\}$ *is the profile set of a user* $i \in U$*. Each profile entry is a* <attribute: profile value> *pair,* $pi_j =< attr_j : pvalue_j >$*;*
- RT *is a set of relationship types supported by the OSN. Each user in an OSN may be connected with others by relationships of different types;*
- RC *is a collection of user relationship sets,* $\{r_1, \dots, r_n\}$*, where* $r_i = \{ri_1, \dots, ri_m\}$ *is the relationship set of a user* $i \in U$*. Each relationship entry is a* <user: relationship type> *pair,* $ri_j =< u_j : rt_j >$*, where* $u_j \in U$ *and* $rt_j \in RT$*;*
- TT *is a set of content types supported by the OSN. Supported content types are photo, video, note, event, status, message, link, and so on;*
- CC *is a collection of user content sets,* $\{c_1, \dots, c_n\}$*, where* $c_i = \{ci_1, \dots, ci_m\}$ *is a set of contents of a user* $i \in U$*. Each content entry is a* <content: content type> *pair,* $ci_j =< cont_j : tt_j >$*, where* $cont_j$ *is a content identifier and* $tt_j \in TT$*;*

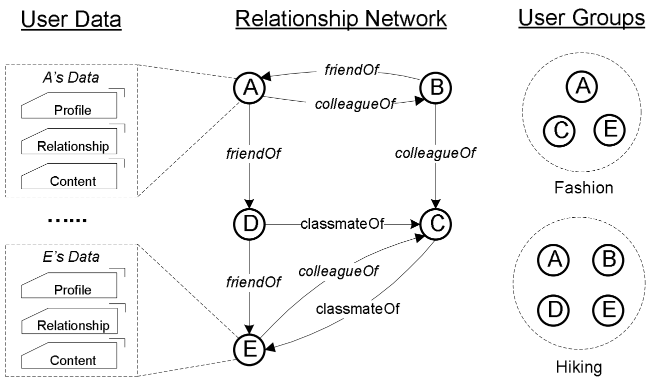


Fig. 2. An Example of Social Network Representation.

- *UU is a collection of uni-directional binary user-to-user relations, $\{UU_{rt_1}, \ldots, UU_{rt_n}\}$, where $UU_{rt_i} \subseteq U \times U$ specifies the pairs of users in a relationship type $rt_i \in RT$; and*
- *$UG \subseteq U \times G$ is a binary user-to-group membership relation;*

Figure 2 shows an example of social network representation. It describes relationships of five individuals, Alice (*A*), Bob (*B*), Carol (*C*), Dave (*D*) and Edward (*E*), along with their groups of interest and their own spaces of data. Note that two users may be directly connected by more than one edge labeled with different relationship types in the relationship network. For example, in Figure 2, Alice (*A*) has a direct relationship of type *colleagueOf* with Bob (*B*), whereas Bob (*B*) has a relationship of *friendOf* with Alice (*A*). Moreover, in this example, we can notice there are two groups that users can participate in: the "*Fashion*" group and the "*Hiking*" group, and some users, such as Alice (*A*) and Edward (*E*), may join in multiple groups.

3.3 Multiparty Authorization Specification

To enable a collaborative authorization management of data sharing in OSNs, it is essential for multiparty access control policies to be in place to regulate access over shared data, representing authorization requirements from multiple associated users. Our policy specification scheme is built upon the above-mentioned OSN model (Section 3.2).

Recently, several access control schemes (e.g., [6, 13, 14]) have been proposed to support fine-grained authorization specifications for OSNs. Unfortunately, these schemes can only allow a single controller (*the resource owner*) to specify access control policies. Indeed, a flexible access control mechanism in a multi-user environment like OSNs is necessary to allow multiple controllers associated with the shared data item to specify access control policies. As we discussed in Section 3.1, in addition to the *owner* of data, other controllers, including the *contributor*, *stakeholder* and *disseminator* of data, also desire to regulate access to the shared data. We formally define these controllers as follows:

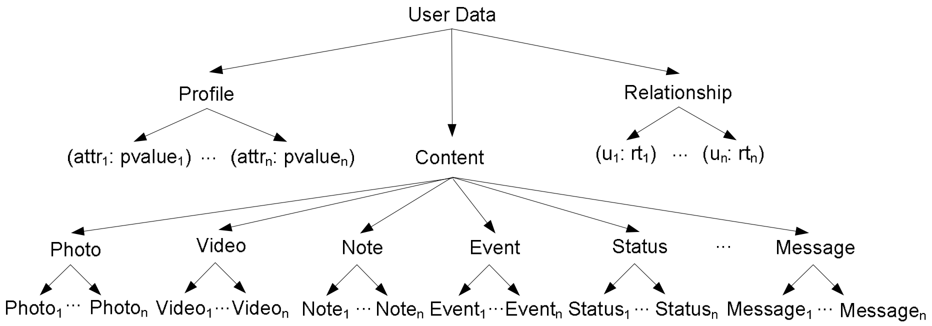


Fig. 3. Hierarchical User Data in OSNs.

Definition 2 *(**Owner**). Let d be a shared data item in the space of a user $i \in U$ in the social network. The user i is called the owner of d, denoted as OW_d^i.*

Definition 3 *(**Contributor**). Let d be a shared data item published by a user $i \in U$ in someone else's space in the social network. The user i is called the contributor of d, denoted as CB_d^i.*

Definition 4 *(**Stakeholder**). Let d be a shared data item published in the space of a user in the social network. Let T be the set of tagged users associated with d. A user $i \in U$ is called a stakeholder of d, denoted as SH_d^i, if $i \in T$.*

Definition 5 *(**Disseminator**). Let d be a shared data item disseminated by a user $i \in U$ from someone else's space to her/his space in the social network. The user i is called a disseminator of d, denoted as DS_d^i.*

In the context of an OSN, user data is composed of three types of information: *User profile* describes who the user is in the OSN, including identity and personal information, such as name, birthday, interests and contact information. *User relationship* shows who the user knows in the OSN, including a list of friends to represent the connections with family members, coworkers, colleagues, and so on. *User content* indicates what the user has in the OSN, including photos, videos, statuses, and all other data objects created through various activities in the OSN. Formally, we define user data as follows:

Definition 6 *(**User Data**). The user data is a collection of data sets, $\{d_1, \ldots, d_n\}$, where $d_i = p_i \cup r_i \cup c_i$ is a set of data of a user $i \in U$ representing the user's profile p_i, the user's relationship list r_i, and the user's content set c_i, respectively.*

User data in OSNs can be organized as a hierarchical structure, whose leaves represent the instances of data, and whose intermediate nodes represent classifications of data. Figure 3 depicts a hierarchical structure of user data where the root node, *user data*, is classified into three types, *profile*, *relationship* and *content*. The content is further divided into multiple categories, such as *photo*, *video*, *note*, *event*, *status*, etc. In this way, access control policies can be specified over both data classifications and instances. Especially, access control policies specified on classifications can be automatically propagated down in the hierarchy. For instance, if access for the parent node *photo* is allowed, access for all children nodes of *photo* is also allowed. As a consequence, such a hierarchical structure of user data can be used to improve the expressiveness of access control policies and simplify the authorization management.

To summarize the aforementioned features and elements, we introduce a formal definition of multiparty access control policies as follows:

Definition 7 ***(Multiparty Access Control Policy)****. A multiparty access control policy is a 7-tuple $P =< controller, ctype, accessor, atype, data, action, effect >$, where*

- *$controller \in U$ is a user who can regulate access to data;*
- *$ctype \in \{OW, CB, SH, DS\}$ is the type of the controller (*owner*,* contributor*,* stakeholder*, and* disseminator*, respectively);*
- *$accessor$ is a set of users to whom the authorization is granted, representing with a set of user names, a set of relationship types or a set of group names. Note that patterns are allowed to specify* any set *by using the the wildcard (*) instead of a specific name;*
- *$atype \in \{UN, RN, GN\}$ is the type of the accessor specification (*user name*,* relationship type*, and* group name*, respectively);*
- *$data \in d_i \cup TT \cup DT$ is a data item $di_j \in d_i$, a content type $tt \in TT$, or a data type $dt \in DT = \{profile, relationship, content\}$, where $i \in U$;*
- *$action = view$ is an action being authorized or forbidden;*[1] *and*
- *$effect \in \{permit, deny\}$ is the authorization effect of the policy.*

Note that different representations of *accessor* in our policy specification scheme have different semantics. If the *accessor* is represented with a set of user names $\{u_1, \ldots, u_n\}$, the semantics of this user name set can be explained as $u_1 \vee \ldots \vee u_n$, which means that any user contained in the user name set is treated as an authorized accessor. On the other hand, if the *accessor* is expressed as a set of relationship types $\{rt_1, \ldots, rt_n\}$ or a set of group names $\{g_1, \ldots, g_n\}$, the semantics of the relationship type set or group name set are interpreted as $rt_1 \wedge \ldots \wedge rt_n$ or $g_1 \wedge \ldots \wedge g_n$. Examples of multiparty access control policies are as follows:

1. $p_1 = (Alice, OW, \{friendOf\}, RN, < statusId, status >, view, permit)$: $Alice$ authorizes her friends to view her status identified by $statusId$. In this policy, $Alice$ is an owner of the status.
2. $p_2 = (Bob, CB, \{colleageOf\}, RN, photo, view, permit)$: Bob authorizes his colleagues to view *all* photos he publishes to others' spaces. In this policy, Bob is a contributor of the photos.
3. $p_3 = (Carol, ST, \{friendOf, colleageOf\}, RN, < photoId, photo >, view, permit)$: $Carol$ authorizes users who are both her friends and her colleagues to view one photo $photoId$ she is tagged in. In this policy, $Carol$ is a stakeholder of the photo.

[1] We limit our consideration to $view$ action. The support of more actions such as *post*, *comment*, *tag*, and *update* does not significantly complicate our approach proposed in this paper.

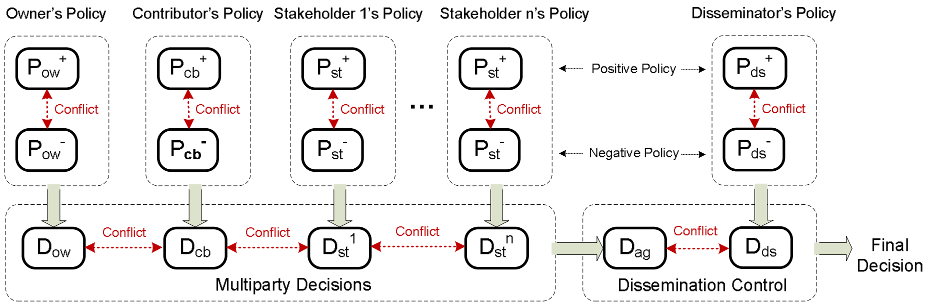


Fig. 4. Conflict Identification for Multiparty Policy Evaluation.

4. $p_4 = (Dave, OW, \{Bob, Carol\}, UN, < eventId, event >, view, deny)$: $Dave$ disallows Bob and $Carol$ to view his event $eventId$.
5. $p_5 = (Edward, DS, \{fashion, hiking\}, GN, < videoId, video >, view, permit)$: $Edward$ authorizes users who are in both groups, $fashion$ and $hiking$, to view a video $videoId$ that he disseminates. In this policy, $Edward$ is a disseminator of the video.

3.4 Multiparty Policy Evaluation

In our proposed multiparty authorization model, each controller can specify a set of policies, which may contains both positive and negative policies, to regulate access of the shared data item. Two steps should be performed to evaluate an access request over multiparty access control policies. The first step checks the access request against policies of each controller and yields a decision for the controller. Bringing in both positive and negative policies in the policy set of a controller raises potential policy conflicts. In the second step, decisions from all controllers responding to the access request are aggregated to make a final decision for the access request. Since those controllers may generate different decisions (`permit` and `deny`) for the access request, conflicts may occurs again. Figure 4 illustrates potential conflicts identified during the evaluation of multiparty access control policies. In order to make an unambiguous final decision for each access request, it is crucial to adopt a systematic conflict resolution mechanism to resolve those identified conflicts during multiparty policy evaluation.

Policy Conflict Resolution in One Party In the first step of multiparty policy evaluation, policies belonging to each controller are evaluated in sequence, and the *accessor* element in a policy decides whether the policy is applicable to a request. If the user who sends the request belongs to the user set derived from the *accessor* of a policy, the policy is applicable and the evaluation process returns a response with the decision (either `permit` or `deny`) indicated by the *effect* element in the policy. Otherwise, the response yields `NotApplicable`. In the context of OSNs, controllers generally utilize a positive policy to define a set of trusted users to whom the shared data item is visible, and a negative policy to exclude some specific untrusted users from whom the shared data item should be hidden. Some general conflict resolution strategies for access control have been introduced [12, 15, 16]. For example, *deny-overrides* (this strategy indicates that "deny" policy take precedence over "allow" policy), *allow-overrides* (this strategy states that "allow" policy take precedence over "deny" policy), *specificity-overrides* (this strategy states a more specific policy overrides more general policies), and *recency-overrides* (this strategy indicates that policies take precedence over policies specified earlier). We can adopt these strategies to resolve policy conflicts in our conflict resolution mechanism when evaluating a controller's policies. Since some strategies, such as *specificity-overrides* and *recency-overrides* are nondeterministic, and *deny-overrides* strategy is too restricted in general for conflict resolution, it is desirable to combine these strategies together to achieve a more effective conflict resolution. Thus, a strategy chain can be constructed to address this issue, which has been discussed in our previous work [17, 18].

Resolving Multiparty Privacy Conflicts When two users disagree on whom the shared data item should be exposed to, we say a *privacy conflict* occurs. The essential reason leading to the privacy conflicts is that multiple controllers of the shared data item often have different privacy concerns over the data item. For example, assume that Alice and Bob are two controllers of a photo. Each of them defines an access control policy stating only her/his friends can view this photo. Since it is almost impossible that Alice and Bob have the same set of friends, privacy conflicts may always exist considering multiparty control over the shared data item.

A *naive* solution for resolving multiparty privacy conflicts is to only allow the common users of accessor sets defined by the multiple controllers to access the data. Unfortunately, this strategy is too restrictive in many cases and may not produce desirable results for resolving multiparty privacy conflicts. Let's consider an example that four users, Alice, Bob, Carol and Dave, are the controllers of a photo, and each of them allows her/his friends to see the photo. Suppose that Alice, Bob and Carol are close friends and have many common friends, but Dave has no common friends with them and also has a pretty weak privacy concern on the photo. In this case, adopting the *naive* solution for conflict resolution may turn out that no one can access this photo. Nevertheless, it is reasonable to give the view permission to the common friends of Alice, Bob and Carol.

A strong conflict resolution strategy may provide a better privacy protection. In the meanwhile, it reduces the social value of data sharing in OSNs. Therefore, it is important to consider the tradeoff between *privacy* and *utility* when resolving privacy conflicts. To address this issue, we introduce a flexible mechanism for resolving multiparty privacy conflicts in OSNs based on a voting scheme. Several simple and intuitive strategies can be derived from the voting scheme as well.

Our voting scheme contains two voting mechanisms, *decision voting* and *sensitivity voting*. In the decision voting, an aggregated decision value from multiple controllers with respect to the results of policy evaluation is computed. In addition, each controller assigns a sensitivity level to the shared data item to reflect her/his privacy concern. Then, a sensitivity score for the data item can be calculated as well through aggregating each controller's sensitivity level value. Based on the aggregated decision value and the sensitivity score, our decision making approach provides two conflict resolution solutions: *automatic* conflict resolution and *strategy-based* conflict resolution. A basic idea of our approach for automatic conflict resolution is that the sensitivity score can be utilized as a *threshold* for decision making. Intuitively, if the sensitivity score is higher, the final decision is likely to *deny* access, taking into account the privacy protection of high sensitive data. Otherwise, the final decision is very likely to *allow* access. Hence, the utility of OSN services cannot be affected. In the second solution, the sensitivity score of a data item is considered as a guideline for the owner of shared data item in selecting an appropriate strategy for conflict resolution. Several specific strategies can be used for resolving multiparty privacy conflicts in OSNs. For example, *owner-overrides* (the owner's decision has the highest priority), *full-consensus-permit* (if any controller denies the access, the final decision is `deny`), *majority-permit* (this strategy permits a request if over 1/2 controllers permit it), *strong-majority-permit* (this strategy permits a request if over 2/3 controllers permit it), and *super-majority-permit* (this strategy permits a request if over 3/4 controllers permit it).

Conflict Resolution for Disseminated Data A user can *share* others' contents with her/his friends in OSNs. In this case, the user is a disseminator of the content, and the content will be posted in the disseminator's space and visible to her/his friends or the public. Since a disseminator may adopt a weaker control over the disseminated content but the content may be much sensitive from the perspective of original controllers of the content, the privacy concerns from the original controllers of the content should be always complied with, preventing inadvertent disclosure of sensitive contents. In other words, the original access control policies should be always enforced to restrict access to the disseminated content. Thus, the final decision for an access request to the disseminated content is a composition of the decisions aggregated from original controllers and the decision from the current disseminator. In order to eliminate the risk of possible leakage of sensitive information from the procedure of data dissemination, we leverage the restrictive conflict resolution strategy, `Deny-overrides`, to resolve conflicts between original controllers' decision and the disseminator's decision. In such a context, if either of those decisions is to `deny` the access request, the final decision is `deny`. Otherwise, if both of them are `permit`, the final decision is `permit`.

4 Prototype Implementation and Evaluation

To demonstrate the feasibility of our authorization model and mechanism, we implemented a Facebook-based application called *MController* for supporting collaborative management of shared data. Our prototype application enables multiple associated users to specify their authorization policies and privacy preferences to co-control a shared data item. We currently restrict our prototype to deal with photo sharing in OSNs. Obversely, our approach can be generalized to handle other kinds of data, such as videos and comments, in OSNs as long as the stakeholders of shared data can be identified with effective methods like tagging or searching.

MController is deployed as a third-party application of Facebook, which is hosted in an Apache Tomcat application server supporting PHP and MySQL database. *MCon-*

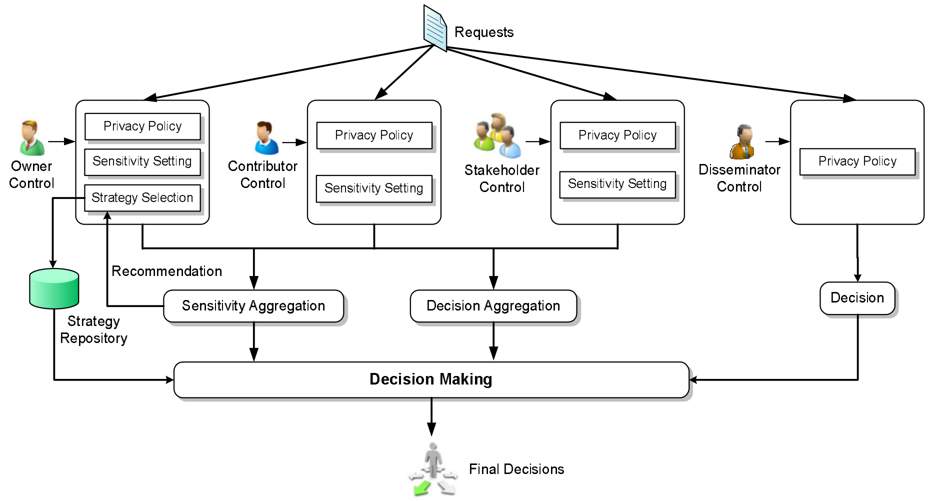
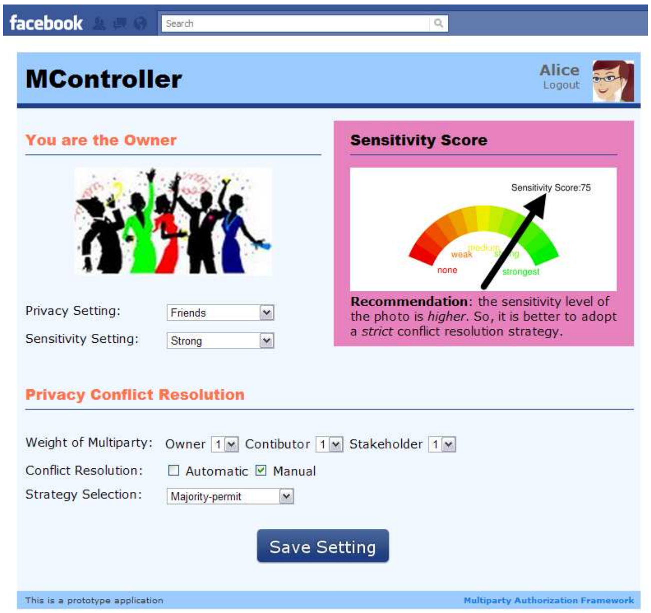


Fig. 5. System Architecture of Decision Making in MController.

troller application is based on the iFrame external application approach, adopting the Facebook REST-based APIs and supporting Facebook Markup Language (FBML), where Facebook server acts as an intermediary between users and the application server. Facebook server accepts inputs from users, then forwards them to the application server. The application server is responsible for the input processing and collaborative management of shared data. Information related to user data such as user identifiers, friend lists, user groups, and user contents are stored in the MySQL database.

Once a user installs *MController* in her/his Facebook space, *MController* can access user's basic information and contents. In particular, *MController* can retrieve and list all photos, which are owned or uploaded by the user, or where the user was tagged. Then, the user can select any photo to define the privacy preference. If the user is not the owner of selected photo, s/he can only edit the privacy setting and sensitivity setting of the photo. Otherwise, if the user is an owner of the photo, s/he can further configure the conflict resolution mechanism for the shared photo.

A core component of *MController* is the decision making module, which processes access requests and returns responses (either `permit` or `deny`) for the requests. Figure 5 depicts a system architecture of the decision making module in *MController*. To evaluate an access request, the policies of each controller of the targeted content are enforced first to generate a decision for the controller. Then, the decisions of all controllers are aggregated to yield a final decision as the response of the request. During the procedure of decision making, policy conflicts are resolved when evaluating controllers' policies by adopting a strategy chain pre-defined by the controllers. In addition, multiparty privacy conflicts are resolved based on the configured conflict resolution mechanism when aggregating the decisions of controllers. If the owner of the content chooses automatic conflict resolution, the aggregated sensitivity value is utilized as a threshold for making a decision. Otherwise, multiparty privacy conflicts are resolved by applying



You are the Owner

Privacy Setting: Friends

Sensitivity Setting: Strong

Sensitivity Score

Recommendation: the sensitivity level of the photo is *higher*. So, it is better to adopt a *strict* conflict resolution strategy.

Privacy Conflict Resolution

Weight of Multiparty: Owner 1 Contibutor 1 Stakeholder 1

Conflict Resolution: □ Automatic ☑ Manual

Strategy Selection: Majority-permit

This is a prototype application

Multiparty Authorization Framework

Fig. 6. MController for Owner Control on Facebook.

the strategy selected by the owner, and the aggregated sensitivity score is considered as a recommendation for the strategy selection. Regarding access requests to the disseminated contents, the final decision is made by combining the disseminator's decision and original controllers' decision through a *deny-overrides* combination strategy.

A snapshot of *MController* for owner control is shown in Figure 6, where an owner of a photo can assign weight values to different types of controllers of the shared photo, and select either *automatic* or *manual* mechanism for conflict resolution. If the owner chooses *manual* conflict resolution, s/he can further select an appropriate conflict resolution strategy referring to the recommendation derived from the sensitivity score of the photo. Note that *MController* currently requires all controllers of a shared photo should define their privacy preferences before applying our authorization mechanism to evaluate the requests. Otherwise, the photo is only visible to the controllers. Since a user may be involved in the control of hundreds of photos, manual input of the privacy preferences is a time-consuming and tedious task. As part of our future work, we would study inference-based techniques [11] for automatically configuring controllers' privacy preferences.

To evaluate the performance of the policy evaluation mechanism in *MController*, we changed the number of the controllers of a shared photo from 1 to 20. Also, we considered two cases for our evaluation. In the first case, each controller has only one positive policy. The second case examines two policies (one positive policy and one negative policy) of each controller. Figure 7 shows the policy evaluation cost while changing the number of the controllers. For both cases, the experimental results show that the policy evaluation cost increased slightly with the increase of the number of the

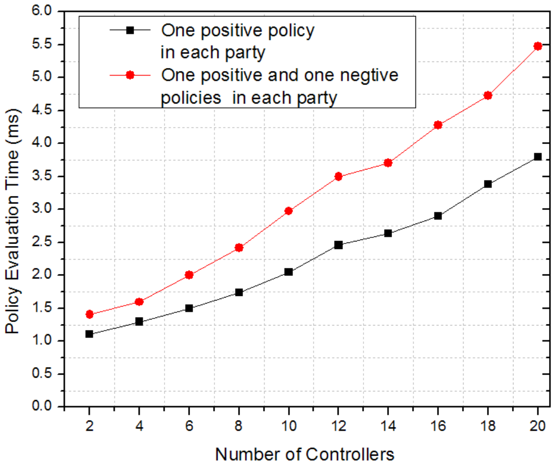


Fig. 7. Performance of Policy Evaluation Mechanism.

controllers. Also, we can observe that *MController* performs fast enough to handle even a large number of controllers for collaboratively managing the shared data.

5 Conclusion and Future Work

In this paper, we have proposed a novel authorization framework that facilitates collaborative management of the shared data in OSNs. We have given an analysis of multiparty authorization requirements in OSNs, and formulated a multiparty access control model. Our access control model is accompanied with a multiparty policy specification scheme and corresponding policy evaluation mechanism. Moreover, we have described a proof-of-concept implementation of our approach called *MController*, which is a Facebook application, along with performance analysis.

As our future work, we will incorporate a logic-based reasoning feature into our approach to provide a variety of analysis services for collaborative management of the shared data. Also, we are planning to conduct extensive user studies to evaluate the usability of our proof-of-concept implementation, *MController*. In addition, as effective automated algorithms (e.g., facial recognition [9, 25]) are being developed to recognize people accurately in contents such as photos and then generate tags automatically, access and privacy controls will become even more problematic in the future. Consequently, we would extend our work to explore more sophisticated and effective solutions to address emerging security and privacy challenges for sharing various data in OSNs.

References

1. Facebook Privacy Policy. `http://www.facebook.com/policy.php/`.
2. Facebook Statistics. `http://http://www.facebook.com/press/info.php?statistics`.

3. Myspace Privacy Policy. `http://www.myspace.com/index.cfm?fuseaction=misc.privacy/`.
4. A. Besmer and H. Richter Lipford. Moving beyond untagging: Photo privacy in a tagged world. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1563–1572. ACM, 2010.
5. S. Brands. *Rethinking public key infrastructures and digital certificates: building in privacy*. The MIT Press, 2000.
6. B. Carminati, E. Ferrari, and A. Perego. Rule-based access control for social networks. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 1734–1744. Springer, 2006.
7. B. Carminati, E. Ferrari, and A. Perego. Enforcing access control in web-based social networks. *ACM Transactions on Information and System Security (TISSEC)*, 13(1):1–38, 2009.
8. E. Carrie. Access Control Requirements for Web 2.0 Security and Privacy. In *Proc. of Workshop on Web 2.0 Security & Privacy (W2SP)*. Citeseer, 2007.
9. J. Choi, W. De Neve, K. Plataniotis, Y. Ro, S. Lee, H. Sohn, H. Yoo, W. Neve, C. Kim, Y. Ro, et al. Collaborative Face Recognition for Improved Face Annotation in Personal Photo Collections Shared on Online Social Networks. *IEEE Transactions on Multimedia*, pages 1–14, 2010.
10. N. Elahi, M. Chowdhury, and J. Noll. Semantic Access Control in Web Based Communities. In *Proceedings of the Third International Multi-Conference on Computing in the Global Information Technology*, pages 131–136. IEEE, 2008.
11. L. Fang and K. LeFevre. Privacy wizards for social networking sites. In *Proceedings of the 19th international conference on World wide web*, pages 351–360. ACM, 2010.
12. K. Fisler, S. Krishnamurthi, L. A. Meyerovich, and M. C. Tschantz. Verification and change-impact analysis of access-control policies. In *ICSE '05: Proceedings of the 27th international conference on Software engineering*, pages 196–205, New York, NY, USA, 2005. ACM.
13. P. Fong. Relationship-Based Access Control: Protection Model and Policy Language. In *Proceedings of the First ACM Conference on Data and Application Security and Privacy*. ACM, 2011.
14. P. Fong, M. Anwar, and Z. Zhao. A privacy preservation model for facebook-style social network systems. In *Proceedings of the 14th European conference on Research in computer security*, pages 303–320. Springer-Verlag, 2009.
15. I. Fundulaki and M. Marx. Specifying access control policies for XML documents with XPath. In *Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 61–69. ACM New York, NY, USA, 2004.
16. S. Jajodia, P. Samarati, and V. S. Subrahmanian. A logical language for expressing authorizations. In *IEEE Symposium on Security and Privacy*, pages 31–42, Oakland, CA, May 1997.
17. J. Jin, G. Ahn, H. Hu, M. Covington, and X. Zhang. Patient-centric authorization framework for sharing electronic health records. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 125–134. ACM, 2009.
18. J. Jin, G. Ahn, H. Hu, M. Covington, and X. Zhang. Patient-centric authorization framework for electronic healthcare services. *Computers & Security*, 30(2-3):116–127, 2011.
19. S. Kruk, S. Grzonkowski, A. Gzella, T. Woroniecki, and H. Choi. D-FOAF: Distributed identity management with access rights delegation. *The Semantic Web–ASWC 2006*, pages 140–154, 2006.
20. N. Li, Q. Wang, W. Qardaji, E. Bertino, P. Rao, J. Lobo, and D. Lin. Access control policy combining: theory meets practice. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 135–144. ACM, 2009.

21. A. Masoumzadeh and J. Joshi. Osnac: An ontology-based access control model for social networking systems. *IEEE International Conference on Privacy, Security, Risk and Trust*, 0:751–759, 2010.
22. M. Shehab, G. Cheek, H. Touati, A. Squicciarini, and P. Cheng. User Centric Policy Management in Online Social Networks. In *2010 IEEE International Symposium on Policies for Distributed Systems and Networks*, pages 9–13. IEEE, 2010.
23. A. Squicciarini, F. Paci, and S. Sundareswaran. PriMa: an effective privacy protection mechanism for social networks. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 320–323. ACM, 2010.
24. A. Squicciarini, M. Shehab, and F. Paci. Collective privacy management in social networks. In *Proceedings of the 18th international conference on World wide web*, pages 521–530. ACM, 2009.
25. Z. Stone, T. Zickler, and T. Darrell. Autotagging Facebook: Social network context improves photo annotation. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.
26. R. Wishart, D. Corapi, S. Marinovic, and M. Sloman. Collaborative Privacy Policy Authoring in a Social Networking Context. In *2010 IEEE International Symposium on Policies for Distributed Systems and Networks*, pages 1–8. IEEE, 2010.
27. G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to de-anonymize social network users. In *2010 IEEE Symposium on Security and Privacy*, pages 223–238. IEEE, 2010.
28. E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th international conference on World wide web*, pages 531–540. ACM, 2009.