ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 9 September 2013 Page No. 2870-2877

User Access Control for Online Social Networks

Narayana Naidu Pachava[#1], Sd. Akthar[#2], P. Babu[#3]

[1]Student (M.Tech), Department of Computer Science & Engineering,
QUBA college of Engineering and Technology, Nellore, A.P. INDIA.
[2]Associate professor, Department of Computer Science & Engineering,
QUBA college of Engineering and Technology, Nellore, A.P. INDIA.
[3]Head of the Department, Department of Computer Science & Engineering,
QUBA college of Engineering and Technology, Nellore, A.P., INDIA.

Abstract *— **In recent years we are experiencing the tremendous growth in Online Social Networks (OSNs) and become a de facto portal for hundreds of millions of Internet users. Digital social interactions and information security are the means offered by these OSNs, but also raise a number of security and privacy issues. In OSNs users are restricted to access the shared data, but they currently do not provide any mechanism to enforce privacy concerns over data associated with multiple users. In our paper, we propose an approach to enable the protection of shared data associated with multiple users in OSNs. We also formulate an access control model to capture the essence of multiparty authorization requirements, along with a multiparty policy specification scheme and a policy enforcement mechanism. To the end, we discuss a proof-of-concept prototype of our approach as part of an application in Facebook and provide usability study and system evaluation of our method.***

Index Terms—Social network, multiparty access control, security model, policy specification and management.

I. INTRODUCTION

Online Social Networks (OSNs) such as Facebook, Google+, and Twitter are inherently designed to enable people to share personal and public information and make social connections with friends, coworkers, colleagues, family and even with strangers. 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 more than 800 million active users and over 30 billion pieces of content (web links, news stories, blog posts, notes, photo albums, etc.) shared each month [3]. To protect user data, access control has become a central feature of OSNs [2], [4].

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 content 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 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*, *groups* or *public* to access their data, depending on their personal authorization and privacy requirements.

Although OSNs currently provide simple access control mechanisms allowing users to govern access to information contained in their own spaces, users, unfortunately, have no 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 a critical issue, preliminary protection mechanisms have been offered by existing OSNs. However, these simple protection mechanisms suffer from several limitations. 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 managing the shared data collaboratively.

In this paper, we pursue a systematic solution to facilitate collaborative management of shared data 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 access control (MPAC) model is 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., [9], [10], [15], [16], [20]). Our model also contains a multiparty policy specification scheme. Meanwhile, since conflicts are inevitable in multiparty authorization enforcement, a voting mechanism is further provided to deal with authorization and privacy conflicts in our model.

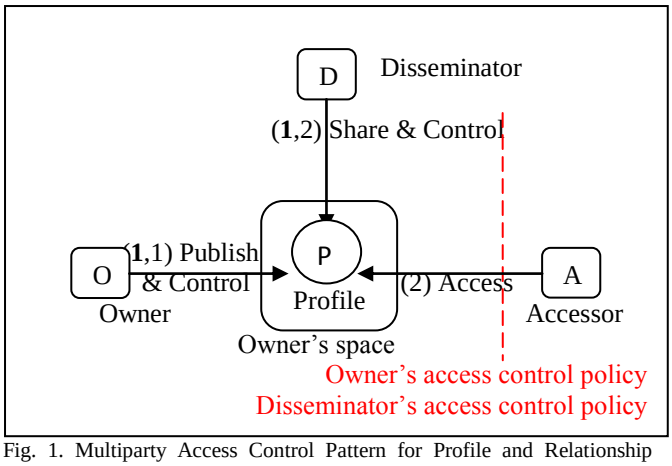

Fig. 1. Multiparty Access Control Pattern for Profile and Relationship Sharing

(a) A disseminator shares other's profile

Another compelling feature of our solution is the support of analysis on multiparty access control model and systems. The correctness of implementation of an access control model is based on the premise that the access control model is valid. Assessing the implications of access control mechanisms traditionally relies on the security analysis technique, which has been applied in several domains (e.g., operating systems [18],). In our approach, we additionally introduce a method to represent and reason about our model in a logic program. Our experimental results demonstrate the feasibility and usability of our approach.

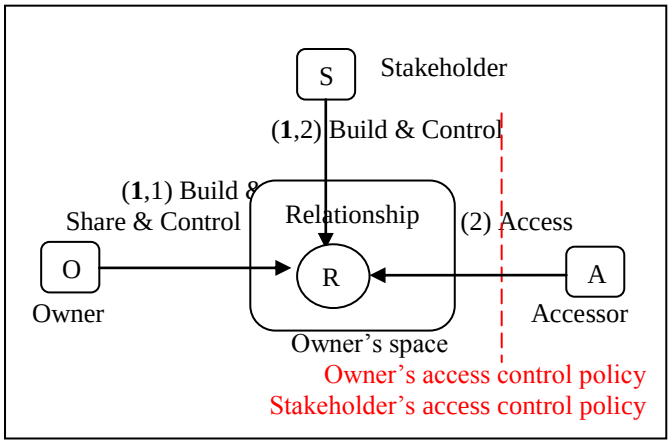

Fig.1. (b) A user shares his/her relationships

The rest of the paper is organized as follows. In Section 2, we present multiparty authorization requirements and access control patterns for OSNs. We articulate our proposed MPAC model, including multiparty authorization specification and multiparty policy evaluation in Section 3 and experimental results are described in Section 4. Section 6 discusses how to 5tackle collusion attacks followed by the related work in Section 6. Section 7 concludes this paper and discusses our future directions.

II. MULTIPARTY ACCESS CONTROL FOR OSNs: REQUIREMENTS AND PATTERNS

In this section we proceed with a comprehensive requirement analysis of multiparty access control in OSNs. Meanwhile, we discuss several typical sharing patterns occurring in OSNs where multiple users may have different authorization requirements to a single resource. We specifically analyze three scenarios — *profile sharing*, *relationship sharing and content sharing* — to understand the risks posted by the lack of collaborative control in OSNs. We leverage Facebook as the running example in our discussion since it is currently the most popular and representative social network provider. In the meantime, we reiterate that our discussion could be easily extended to other existing social network platforms, such as Google+ [6].

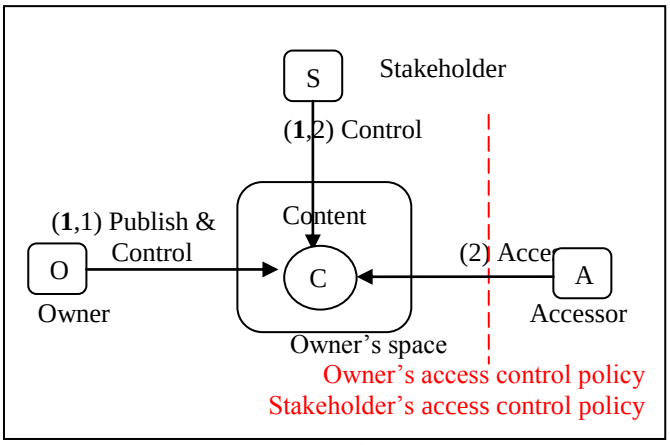

Fig.2. Multiparty Access Control Pattern for Content Sharing

(a) A shared content has multiple stakeholders

Profile sharing: An appealing feature of some OSNs is to support *social applications* written by third-party developers to create additional functionalities built on the top of users' profile for OSNs [1], [5]. To provide meaningful and attractive services, these social applications consume user profile attributes, such as name, birthday, activities, interests, and so on. To make matters more complicated, social applications on current OSN platforms can also consume the profile attributes of a user's friends. In this case, users can select particular pieces of profile attributes they are willing to share with the applications when their friends use the applications. At the same time, the users who are using the applications may also want to control what information of their friends is available to the applications since it is possible for the applications to infer their private profile attributes through their friends' profile attributes [23], [27]. This means that when an application accesses the profile attributes of a user's friend, both the user and her friend want to gain control over the profile attributes. If we consider the application is an *accessor*, the user is a *disseminator* and the user's friend is the *owner* of shared profile attributes in this scenario, Figure 1(a) demonstrates a profile sharing pattern where a disseminator can share others' profile attributes to an accessor. Both the owner and the disseminator can specify access control policies to restrict the sharing of profile attributes.

Relationship sharing: Another feature of OSNs is that users can share their relationships with other members. Relationships are inherently bidirectional and carry potentially sensitive information that associated users may not want to disclose. Most OSNs provide mechanisms that users can regulate the display of their friend lists. A user, however, can only control one direction of a relationship. Figure 1(b) shows a relationship sharing pattern where a user called *owner*, who has a relationship with another user called *stakeholder*, shares the relationship with an *accessor*. In this scenario, authorization requirements from both the

owner and the stakeholder should be considered. Otherwise, the stakeholder's privacy concern may be violated.

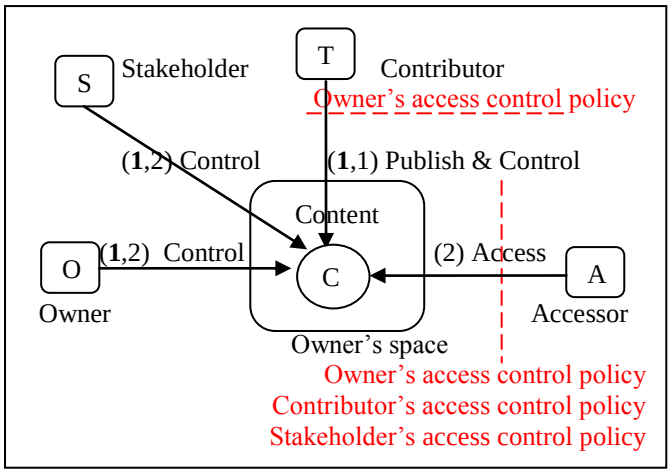

Fig.2. (b) A shared content is published by a contributor

Content sharing: OSNs provide built-in mechanisms enabling users to communicate and share contents with other members. OSN users can post statuses and notes, upload photos and videos in their own spaces, 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. Figure 2(b) shows a content sharing pattern reflecting this scenario where a contributor publishes content to other's space and the content may also have multiple stakeholders (e.g., tagged users). All associated users should be allowed to define access control policies for the shared content.

III. MULTIPARTY ACCESS CONTROL MODEL FOR OSNs

In this section, we formalize a MultiParty Access Control (MPAC) model for OSNs (Section 3.1), as well as a policy scheme (Section 3.2) and a policy evaluation mechanism (Section 3.3) for the specification and enforcement of MPAC policies in OSNs.

3.1 MPAC Model

Recently, several access control schemes (e.g., [9], [10], [15], [16]) 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 should allow multiple controllers, who are associated with the shared data, to specify access control policies. As we identified previously in the sharing patterns (Section 2), in addition to the *owner* of data, other controllers, including the *contributor*, *stakeholder* and *disseminator* of data, need to regulate the access of the shared data as well. We define these controllers as follows:

Definition 1: (Owner). Let *d* be a data item in the space of a user *u* in the social network. The user *u* is called the owner of *d*.

Definition 2: (Contributor). Let *d* be a data item published by a user *u* in someone else's space in the social network. The user *u* is called the contributor of *d*.

Definition 3: (Stakeholder). Let *d* be a data item in the space of a user in the social network. Let *T* be the set of tagged users associated with *d*. A user *u* is called a stakeholder of *d*, if *u 2 T*.

Definition 4: (Disseminator). Let *d* be a data item shared by a user *u* from someone else's space to his/her space in the social network. The user *u* is called a disseminator of *d*.

We now formally define our MPAC model as follows:

- $U = \{u_1, \ldots, u_n\}$ is a set of users of the OSN. Each user has a unique identifier;
- $G = \{g_1, \ldots, g_n\}$ is a set of groups to which the users can belong. Each group also has a unique identifier;
- $P = \{p_1, \ldots, p_n\}$ is a collection of user profile sets, where $pi = \{q_{i1}, \ldots, q_{im}\}$ is the profile of a user $i \in U$. Each profile entry is a <*attribute: profile-value*> pair, $q_{ij} =< attr_j : pvalue_j >$, where $attr_j$ is an attribute identifier and $pvalue_j$ is the attribute value;
- *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;
- $R = \{r_1; : : : ; r_n\}$ is a collection of user relationship sets, where $r_i = \{s_{i1}; : : : ; s_{im}\}$ is the relationship list of a user $i \in U$. Each relationship entry is a <*user: relationship-type*> pair, $s_{ij} =< u_j : rt_j >$, where $u_j \in U$, $rtj \in RT$;
- $C = \{c_1; : : : ; c_n\}$ is a collection of user content sets, where $ci = \{e_{i1}; : : : ; e_{im}\}$ is a set of contents of a user $i \in U$, where e_{ij} is a content identifier;
- $D = \{d_1; : : : ; d_n\}$ is a collection of data sets, where $d_i = p_i \cup r_i \cup c_i$ is a set of data of a user $i \in U$;
- *CT* = {*OW,CB,ST,DS*} is a set of controller types, indicating *ownerOf*, *contributorOf*, *stakeholderOf*, and *disseminatorOf*, respectively;
- $UU = \{UU_{rt1}; \ldots, UU_{rtn}\}$ is a collection of uni-directional binary user-to-user relations, where UU_{rti} $U * U$ specifies the pairs of users having relationship type *rti* $\in RT$;
- $UG \_ U * G$ is a set of binary user-to-group membership relations;
- $UD = \{UD_{ct1}, \ldots, UD_{ctn}\}$ is a collection of binary user-to-data relations, where $UD_{cti} \_ U*D$ specifies a set of < *user, data* > pairs having controller type $ct_i \in CT$;
- *relation members* : $U \rightarrow^{RT} 2^U$, a function mapping each user $u \in U$ to a set of users with whom s/he has a relationship $rt \in RT$:
  relation members$(u : U; rt : RT) = \{u' \in U \mid (u, u') \in sUU_{rt}\}$;
- *ROR members*: $U \rightarrow^{RT} 2^U$, a function mapping each user $u \in U$ to a set of users with whom s/he has a *transitive* relation of a relationship $rt \in RT$, denoted as *relationships-of-relationships* (ROR).:
  ROR members$(u : U, rt : RT) = \{u' \in U \mid u' \in$ *relation members*$(u, rt) \vee (\exists u'' \in U \mid u'' \in$ *ROR members*$(u, rt)) \wedge u' \in$ *ROR members*$(u''; rt)]\}$;
- *controllers* : $D \rightarrow^{CT} 2^U$; a function mapping each date item $d \in D$ to a set of users who are the controller with the controller type $ct \in CT$:
  controllers$(d : D, ct : CT) = \{u \in U \mid (u, d) \in UD_{ct}\}$; and
- *group members* : $G \rightarrow 2^U$; a function mapping each group $g \in G$ to a set of users who belong to the group: *group members*$(g : G) = \{u \in U \mid (u; g) \in UG\}$; *groups*$(u : U) = \{g \in G \mid (u; g) \in UG\}$;

3.2 MPAC Policy 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 proposed MPAC model.

Accessor Specification: Accessors are a set of users who are granted to access the shared data. Accessors can be represented with a set of user names, a set of relationship names or a set of group names in OSNs.

Data Specification: In OSNs, user data is composed of three types of information, *user profile*, *user relationship* and *user content*. To facilitate effective privacy conflict resolution for multiparty access control, we introduce *sensitivity levels* for data specification, which are assigned by the controllers to the shared data items. A user's judgment of the sensitivity level of the data is not binary (private/public), but multi-dimensional with varying degrees of sensitivity.

Access Control Policy: To summarize the above-mentioned policy elements, we introduce the definition of a multiparty access control policy as follows:

Definition 6: (MPAC Policy). A multiparty access control policy is a 5-tuple *P =< controller; ctype; accessor; data; effect >*, where

- *controller 2 U* is a user who can regulate the access of *data*;
- *ctype 2 CT* is the type of the controller;
- *accessor* is a set of users to whom the authorization is granted, representing with an access specification defined in Definition 5.
- *data* is represented with a data; and
- *effect 2 fpermit; denyg* is the authorization effect of the policy.

A controller can leverage five sensitivity levels: 0.00 (*none*), 0.25 (*low*), 0.50 (*medium*), 0.75 (*high*), and 1.00 (*highest*) for the shared data

3.3 Multiparty Policy Evaluation

Two steps are performed to evaluate an access request over multiparty access control policies. The first step checks the access request against the policy specified by each controller and yields a decision for the controller. 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 deny decision if the policy is not applicable to the request. In the second step, decisions from all controllers responding to the access request are aggregated to make a final decision for the access request. Figure 4 illustrates the evaluation process of multiparty access control policies. Since data controllers may generate different decisions (permit and deny) for an access request, *conflicts* may occur. In order to make an unambiguous decision for each access request, it is essential to adopt a systematic conflict resolution mechanism to resolve those conflicts during multiparty policy evaluation. The essential reason leading to the conflicts – especially *privacy* conflicts.

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 item. Unfortunately, this strategy is too restrictive in many cases and may not produce desirable results for resolving multiparty privacy conflicts. A *strong* conflict resolution strategy may provide a better privacy protection. Meanwhile, it may reduce 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 simple but flexible voting scheme for resolving multiparty privacy conflicts in OSNs.

3.3.1 A voting scheme for decision making of multiparty control.

Majority voting is a popular mechanism for decision making [21]. Inspired by such a decision making mechanism, we propose a voting scheme to achieve an effective multiparty conflict resolution for OSNs. A notable feature of the voting mechanism for conflict resolution is that the decision from each controller is able to have an effect on the final decision. Our voting scheme contains two voting mechanisms, *decision voting* and *sensitivity voting*.

Decision Voting: A decision voting value (*DV*) derived from the policy evaluation is defined as follows, where *Evaluation*(*p*) returns the decision of a policy *p*:

$$DV = \{ 0 \text{ if } Evaluation(p) = Deny\ 1; \text{ if } Evaluation(p) = Permit \quad (1)$$

Assume that all controllers are equally important, an aggregated decision value (DV_{ag}) (with a range of 0.00 to 1.00) from multiple controllers including the owner (DV_{ow}), the contributor (DV_{cb}) and stakeholders (DV_{st}), is computed with following equation:

$$DV_{ag} = (DV_{ow} + DV_{cb} + \sum_{i \in SS} DV^{i}_{st}) * 1/m \quad (2)$$

where *SS* is the stakeholder set of the shared data item, and *m* is the number of controllers of the shared data item.

Each controller of the shared data item may have (i) a different trust level over the data owner and (ii) a different reputation value in terms of collaborative control. Thus, a generalized decision voting scheme needs to introduce weights, which can be calculated by aggregating trust levels and reputation values [17], on different controllers. Different weights of controllers are essentially represented by different importance degrees on the aggregated decision. In general, the importance degree of controller *x* is "weight$_x$ / sum of weights". Suppose that $!_{ow}$, $!_{cb}$ and $!_{i\ sh}$ are weight values for owner, contributor and stakeholders, respectively, and *n* is the number of stakeholders of the shared data item.A weighted decision voting scheme is as follows:

$$DV_{ag} = (!_{ow} \_ DV_{ow} + !_{cb} \_ DV_{cb} + \Sigma_{n\ i=1} (!_{i\ st} \_ DV_{I\ st})) \_ 1\ !_{ow} + !_{cb} + \Sigma_{n\ i=1}\ !_{i\ st} \quad (3)$$

Sensitivity Voting: Each controller assigns a sensitivity level (*SL*) to the shared data item to reflect her/his privacy concern. A sensitivity score (SC) (in the range from 0.00 to 1.00) for the data item can be calculated based on following equation:

$$SC = (SL_{ow} + SL_{cb} + \Sigma_{i2SS} SL_{i\ st}) \_ 1\ m \quad (4)$$

Note that we can also use a generalized sensitivity voting scheme like equation (3) to compute the sensitivity score (SC).

3.3.2 Threshold-based conflict resolution

A basic idea of our approach for threshold-based conflict resolution is that the sensitivity score (SC) can be utilized as a *threshold* for decision making. Intuitively, if the sensitivity score is higher, the final decision has a high chance to *deny* access, taking into account the privacy protection of high sensitive data. Otherwise, the final decision is very likely to *allow* access, so that the utility of OSN services cannot be affected. The final decision is made automatically by OSN systems with this threshold-based conflict resolution as follows:

$$Decision = \{ \text{Permit if } DV_{ag} > SC \text{ Deny if } DV_{ag} \_ SC \quad (5)$$

It is worth noticing that our conflict resolution approach has an *adaptive* feature which reflects the changes of policies and sensitivity levels. If any controller changes her/his policy or sensitivity level for the shared data item, the aggregated decision value (DV_{ag}) and the sensitivity score (SC) will be recomputed and the final decision may be changed accordingly.

3.3.3 Strategy-based conflict resolution with privacy recommendation

In this conflict resolution, the sensitivity score (SC) of a data item is considered as a guideline for the owner of shared data item in selecting an appropriate strategy for conflict resolution. We introduce following strategies for the purpose of resolving multiparty privacy conflicts in OSNs.

- Owner-overrides: the owner's decision has the highest priority. This strategy achieves the owner control mechanism that most OSNs are currently utilizing for data sharing. Based on the weighted decision voting scheme, we set $!_{ow} = 1$, $!_{cb} = 0$ and $!_{st} = 0{,}1$ and the final decision can be made as follows:
  $$Decision = \{ \text{Permit if } DV_{ag} = 1 \text{ Deny if } DV_{ag} = 0 \quad (6)$$
- Full-consensus-permit: if any controller denies the access, the final decision is deny. This strategy can achieve the *naive* conflict resolution that we discussed previously. The final decision can be derived as:
  $$Decision = \{ \text{Permit if } DV_{ag} = 1 \text{ Deny otherwise} \quad (7)$$
- Majority-permit: this strategy permits (denies, resp.) a request if the number of controllers to permit (deny, resp.) the request is greater than the number of controllers to deny (permit, resp.) the request. The final decision can be made as:
  $$Decision = \{ \text{Permit if } DV_{ag} \_ 1{=}2 \text{ Deny if } DV_{ag} < 1{=}2 \quad (8)$$

Other majority voting strategies [22] can be easily supported by our voting scheme, such as *strong-majoritypermit* (this strategy permits a request if over 2/3 controllers permit it), *super-majority-permit* (this strategy permits a request if over 3/4 controllers permit it).

3.3.4 Conflict resolution for dissemination control

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 more sensitive from the perspective of original controllers of the content, the privacy concerns from the original controllers of the content should be always fulfilled, 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 a 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.

IV. IMPLEMENTATION AND EVALUATION

4.1 Prototype Implementation

We implemented a proof-of-concept Facebook application for the collaborative management of shared data, called *MController* (http://apps.facebook.com/MController). Our prototype application enables multiple associated users to specify their authorization policies and privacy preferences to co-control a shared data item. It is worth noting that our current implementation was restricted to handle photo sharing in OSNs. Obviously, our approach can be generalized to deal with other kinds of data sharing, such as videos and comments, in OSNs as long as the stakeholder of shared data are identified with effective methods like tagging or searching.

The architecture of *MController*, which is divided into two major pieces, *Facebook server* and *application server*. The Facebook server provides an entry point via the Facebook application page, and provides references to photos, friendships, and feed data through API calls. 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 application database. Users can access the *MController* application through Facebook, which serves the application in an iFrame. When access requests are made to the decision making portion in the application server, results are returned in the form of access to photos or proper information about access to photos. In addition, when privacy changes are made, the decision making portion returns change-impact information to the interface to alert the user. Moreover, analysis services in *MController* application are provided by implementing an ASP translator, which communicates with an ASP reasoner. Users can leverage the analysis services to perform complicated authorization queries.

MController is developed as a third-party Facebook application, which is hosted in an Apache Tomcat application server supporting PHP and MySQL database. A core component of *MController* is the decision making module, which processes access requests and returns responses (either permit or deny) for the requests. In 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. 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 decision making. Otherwise, multiparty privacy conflicts are resolved by applying the strategy selected by the owner, and the aggregated sensitivity score is considered as a recommendation for strategy selection. Regarding the access requests to disseminated content, the final decision is made by combining the disseminator's decision and original controllers' decision adopting corresponding combination strategy discussed previously.

By default, the conflict resolution is set to automatic. However, if the owner chooses to set a manual conflict resolution, s/he is informed of a sensitivity score of shared photo and receives a recommendation for choosing an appropriate conflict resolution strategy. Once a controller saves her/his privacy setting, a corresponding feedback is provided to indicate the potential authorization impact of her/his choice. The controller can immediately determine how many users can see the photo and should be denied, and how many users cannot see the photo and should be allowed. MController can also display the details of all users who violate against the controller's privacy setting. The purpose of such feedback information is to guide the controller to evaluate the impact of collaborative authorization. If the controller is not satisfied with the current privacy control, s/he may adjust her/his privacy setting, contact the owner of the photo to ask her/him to change the conflict resolution strategies, or even report a privacy violation to OSN administrators who can delete the photo. A controller can also perform authorization analysis by advanced queries. Both *over-sharing* and *under-sharing* can be examined by using such an analysis service in *MController*.

4.2 System Usability and Performance Evaluation

4.2.1 Participants and Procedure

MController is a functional proof-of-concept implementation of collaborative privacy management. To measure the practicality and usability of our mechanism, we conducted a survey study (n=35) to explore the factors surrounding users' desires for privacy and discover how we might improve those implemented in *MController*. Specifically, we were interested in users' perspectives on the current Facebook privacy system and their desires for more control over photos they do not own. We recruited participants through university mailing lists and through Facebook itself using Facebook's built-in sharing API. Users were given the opportunity to share our application and play with their friends. While this is not a random sampling, recruiting using the natural dissemination features of Facebook arguably gives an accurate profile of the ecosystem.

Participants were first asked to answer some questions about their usage and perception of Facebook's privacy controls, then were invited to watch a video (http://bit.ly/MController) describing the concept behind *MController*. Users were then instructed to install the application using their Facebook profiles and complete the following actions: set privacy settings for a photo they do not own but are tagged in, set privacy settings for a photo they own, set privacy settings for a photo they contributed, and set privacy settings for a photo they disseminated. As users completed these actions, they answered questions on the usability of the controls in *MController*. Afterward, they were asked to answer further questions and compare their experience with *MController* to that in Facebook.

4.2.2 User Study of MController

For evaluation purposes, questions (http://goo.gl/eDkaV) were split into three areas: *likeability*, *simplicity*, and *control*. *Likeability* is a measure of a user's satisfaction with a system. *Simplicity* is a measure how intuitive and useful the system is. *Control* is a measure of the user's perceived control of their personal data. Questions were either True/False or measured on a 5-point likert scale, and all responses were scaled from 0 to 1 for numerical analysis. In the measurement, a higher number indicates a positive perception or opinion of the system while a lower number indicates a negative one. To analyze the average user perception of the system, we used a 95% confidence interval for the users' answers. This assumes the population to be mostly normal.

Metric	Facebook		MController	
	Average	Upper bound on 95% confidence interval	Average	Lower bound on 95% confidence interval
Likability	0.20	0.25	0.83	0.80
Simplicity	0.38	0.44	0.72	0.64
Control	0.20	0.25	0.83	0.80

Table 1. Usability Evaluation for Facebook and MController Privacy Controls.

Before Using MController. Since we were interested in the maximum average perception of Facebook, we looked at the upper bound of the confidence interval.

An average user asserts at most 25% positively about the *likability* and *control* of Facebook's privacy management mechanism, and at most 44% on Facebook's *simplicity* as shown in Table 1. This demonstrates an average negative opinion of the Facebook's privacy controls that users currently must use.

After Using MController. Users were then asked to perform a few tasks in *MController*. Since we were interested in the average minimum opinion of *MController*, we looked at the lower bound of the confidence interval.

An average user asserts at least 80% positively about the *likability* and *control*, and at least 67% positively on *MController*'s *simplicity* as shown in Table 1. This demonstrates an average positive opinion of the controls and ideas presented to users in *MController*.

4.2.3 Performance Evaluation

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, and assigned each controller with the average number of friends, 130, which is claimed by Facebook statistics [3]. Also, we considered two cases for our evaluation. In the first case,

each controller allows "friends" to access the shared photo. In the second case, controllers specify "friends of friends" as the accessors instead of "friends". In our experiments, we performed 1,000 independent trials and measured the performance of each trial. Since the system performance depends on other processes running at the time of measurement, we had initial discrepancies in our performance. To minimize such an impact, we performed 10 independent trials (a total of 10,000 calculations for each number of controllers). For both cases, the experimental results showed that the policy evaluation time increases linearly with the increase of the number of controllers. With the simplest implementation of our mechanism, where n is the number of controllers of a shared photo, a series of operations essentially takes place n times. There are O(n) MySQL calls and data fetching operations and O(1) for additional operations. Moreover, we could observe there was no significant overhead when we run *MController* in Facebook.

V. DISCUSSIONS

In our multiparty access control system, a group of users could collude with one another so as to manipulate the final access control decision. Consider an attack scenario, where a set of malicious users may want to make a shared photo available to a wider audience. Suppose they can access the photo, and then they all tag themselves or fake their identities to the photo. In addition, they collude with each other to assign a very low sensitivity level for the photo and specify policies to grant a wider audience to access the photo. With a large number of colluding users, the photo may be disclosed to those users who are not expected to gain the access. To prevent such an attack scenario from occurring, three conditions need to be satisfied: (1) there is no fake identity in OSNs; (2) all tagged users are real users appeared in the photo; and (3) all controllers of the photo are honest to specify their privacy preferences.

Regarding the first condition, two typical attacks, Sybil attacks [13] and Identity Clone attacks [8], have been introduced to OSNs and several effective approaches have been recently proposed to prevent the former [14], [26] and latter attacks [19], respectively. To guarantee the second condition, an effective tag validation mechanism is needed to verify each tagged user against the photo. In our current system, if any users tag themselves or others in a photo, the photo owner will receive a tag notification. Then, the owner can verify the correctness of the tagged users. As effective automated algorithms (e.g., facial recognition [12]) are being developed to recognize people accurately in contents such as photos, automatic tag validation is feasible. Considering the third condition, our current system provides a function to indicate the potential authorization impact with respect to a controller's privacy preference. Using such a function, the photo owner can examine all users who are granted the access by the collaborative authorization and are not explicitly granted by the owner her/himself. Thus, it enables the owner to discover potential malicious activities in collaborative control. The detection of collusion behaviors in collaborative systems has been addressed by the recent work [24], [25]. Our future work would integrate an effective collusion detection technique into MPAC. To prevent collusion activities, our current prototype has implemented a function for owner control, where the photo owner can disable any controller, who is suspected to be malicious, from participating in collaborative control of the photo. In addition, we would further investigate how users' reputations–based on their collaboration activities– can be applied to prevent and detect malicious activities in our future work.

VI. RELATED WORK

Access control for OSNs is still a relatively new research area. Several access control models for OSNs have been introduced (e.g., [9], [10], [15], [16], [20]). 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 [20] is primarily a Friend of a Friend (FOAF) ontology-based distributed identity management system for OSNs which indicates the level of friendship between the users participating in a given relationship. Fong et al. [16] 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 [11] described relationship-based access control as one of new security paradigms that addresses unique requirements of Web 2.0. Then, Fong [15] 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 works could model and analyze access control requirements with respect to collaborative authorization management of shared data in OSNs.

In 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. In particular, our proposed solution can also conduct various analysis tasks on access control mechanisms used in OSNs, which has not been addressed by prior work.

VII. CONCLUSION

In this paper, we have proposed a novel solution for collaborative management of shared data in OSNs. A multiparty access control model was formulated, along with a multiparty policy specification scheme and corresponding policy evaluation mechanism. In addition, we have introduced an approach for representing and reasoning about our proposed model. A proof-of-concept implementation of our solution called *MController* has been discussed as well, followed by the usability study and system evaluation of our method.

As part of future work, we are planning to investigate more comprehensive privacy conflict resolution approach and analysis services for collaborative management of shared data in OSNs. Also, we would explore more criteria to evaluate the features of our proposed MPAC model. Besides, we plan to systematically integrate the notion of trust and reputation into our MPAC model and investigate a comprehensive solution to cope with collusion attacks for providing a robust MPAC service in OSNs.

REFERENCES

[1] Facebook Developers. http://developers.facebook.com/.
[2] Facebook Privacy Policy. http://www.facebook.com/policy.php/.
[3]Facebook Statistics. http://www.facebook.com/press/info.php? statistics.
[4]Google+ Privacy Policy. http://http://www.google.com/intl/en/+/ policy/.
[5]OpenSocial Framework. http://code.google.com/p/ opensocialresources/.
[6] The Google+ Project. https://plus.google.com.
[8] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, pages 551–560. ACM, 2009.
[9] 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.
[10] 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.
[11] 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.
[12] J. Choi, W. De Neve, K. Plataniotis, and Y. Ro. Collaborative face recognition for improved face annotation in personal photo collections shared on online social networks. *Multimedia, IEEE Transactions on*, 13(1):14–28, 2011.
[13] J. Douceur. The sybil attack. *Peer-to-peer Systems*, pages 251–260, 2002.
[14] P. Fong. Preventing sybil attacks by privilege attenuation: A design principle for social network systems. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 263–278. IEEE, 2011.
[15] 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*, pages 191–202. ACM, 2011.
[16] 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.
[17] J. Golbeck. Computing and applying trust in web-based social networks. Ph.D. thesis, University of Maryland at College Park College Park, MD, USA. 2005.
[18] M. Harrison, W. Ruzzo, and J. Ullman. Protection in operating systems. *Communications of the ACM*, 19(8):461–471, 1976.
[19] L. Jin, H. Takabi, and J. Joshi. Towards active detection of identity clone attacks on online social networks. In *Proceedings of the first ACM conference on Data and application security and privacy*, pages 27–38. ACM, 2011.
[20] 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.
[21] L. Lam and S. Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 27(5):553–568, 2002.
[22] 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.
[23] A. Mislove, B. Viswanath, K. Gummadi, and P. Druschel. You are who you know: Inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 251–260. ACM, 2010.
[24] B. Qureshi, G. Min, and D. Kouvatsos. Collusion detection and prevention with fire+ trust and reputation model. In *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pages 2548–2555. IEEE, 2010.
[25] E. Staab and T. Engel. Collusion detection for grid computing. In *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 412–419. IEEE Computer Society, 2009.
[26] B. Viswanath, A. Post, K. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. In *ACM SIGCOMM Computer Communication Review*, volume 40, pages 363–374. ACM, 2010.
[27] 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.