International Journal of Academic Research
ISSN: 2348-7666 Vol.2, Issue-1(4), January-March, 2015

Multiparty Access Control for Online Social Networks: Model and Mechanisms

[1]Pitta Venkatesh, M.Tech (Final Year), [2]Department CSE, Baba Institute of Technology and Sciences (BITS), PM Palem. Madhurawada,Visakhapatnam, A.P, India

Mr. Reddi Prasad, Assistant Professor, Department CSE, Baba Institute of Technology and Sciences (BITS), PM Palem. Madhurawada,Visakhapatnam, A.P, India

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 any mechanism to enforce privacy concerns over data associated with multiple users. To this end, we propose an approach to enable the protection of shared data associated with multiple users in OSNs. We 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. Besides, we present a logical representation of our access control model that allows us to leverage the features of existing logic solvers to perform various analysis tasks on our model. We also discuss a proof-of-concept prototype of our approach as part of an application in Face book and provide usability study and system evaluation of our method.

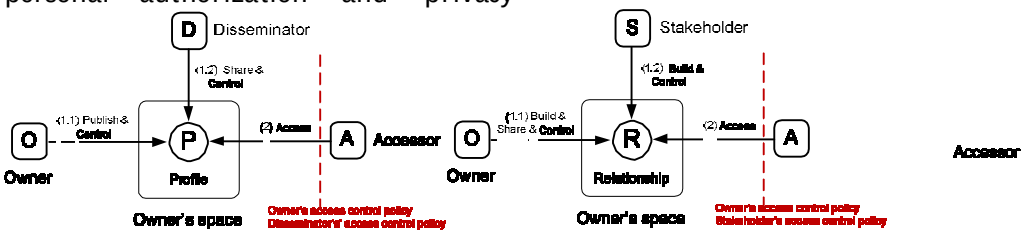
Key words: *Social network, multiparty access control, security model, policy specification and management*

1 Introduction

Online social networks (OSNs) such as Facebook, Googleþ, 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, and so on.) 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 webpages, 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 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* (FOF), *groups*, or *public* to access their data, depending on their personal authorization and privacy requirements.

It is essential to develop an effective and flexible access control mechanism for OSNs, accommodating the special authorization requirements coming from multiple asso- ciated users for managing the shared data collaboratively.



(a) A disseminator shares other's profile (b) A user shares her/his relationships

Fig. 1. MPAC pattern for profile and relationship sharing.

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 (MPAC) for data sharing in OSNs can undermine the protection of user data. Some typical data sharing patterns with respect to multiparty authorization in OSNs are also identified. Based on these sharing patterns, an MPAC model is formulated to capture the core features of multiparty authorization requirements that have not been accommodated so far by existing access control systems and models for OSNs (e.g., [9], [10], [14], [15], [20]). Our model also contains a multiparty policy specification scheme. Meanwhile, since conflicts are inevitable in multi- party authorization enforcement, a voting mechanism is further provided to deal with authorization and privacy conflicts in our model.

Another compelling feature of our solution is the support of analysis on the MPAC model and systems. The correctness of implementation of an access control model is based on the premise that the access control model is valid. Moreover, while the use of an MPAC mechanism can greatly enhance the flexibility for regulating data sharing in OSNs, it may potentially reduce the certainty of system authorization consequences due to the reason that author- ization and privacy conflicts need to be resolved elegantly. 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 [16], trust management [21], and role-based access control [6], [17]). In our approach, we additionally introduce a method to represent and reason about our model in a logic program. In addition, we provide a prototype implementa- tion of our authorization mechanism in the

context of Facebook. Our experimental results demonstrate the feasi- bility and usability of our approach.

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. Section 4 addresses the logical representation and analysis of MPAC. The details about prototype implementation and experimental results are described in Section 5. Section 6 discusses how to tackle collusion attacks followed by the related work in Section 7. Section 8 concludes this paper and discusses our future directions.

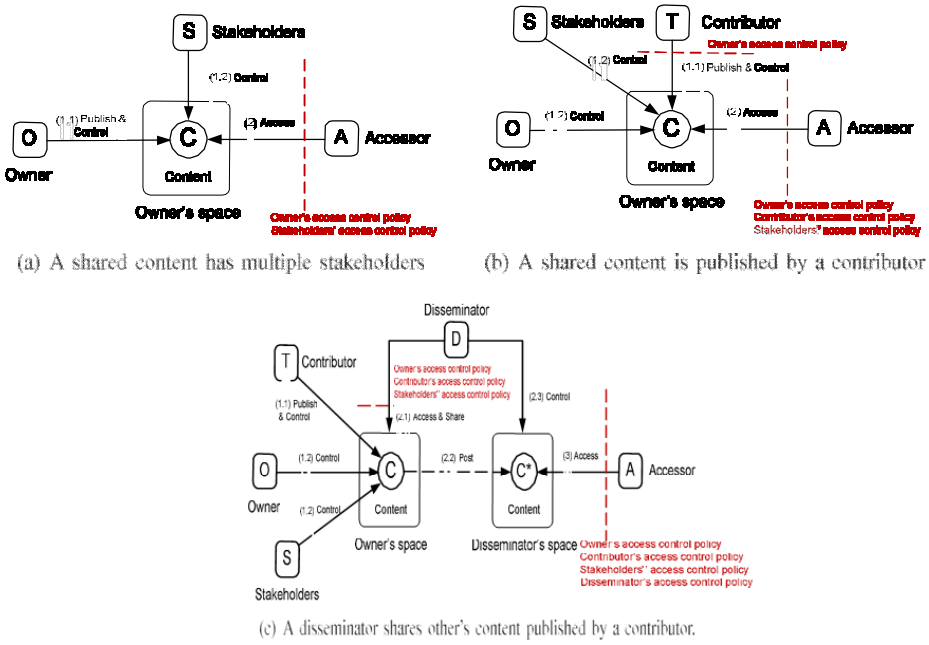
2 MPAC FOR OSNs: Requirements and Patterns

In this section, we proceed with a comprehensive require- ment analysis of MPAC in OSNs. Meanwhile, we discuss several typical sharing patterns occurring in OSNs where multiple users may have different authorization require- ments to a single resource. We specifically analyze three scenarios—profile sharing, relationship sharing, and con- tent sharing—to understand the risks posted by the lack of collaborative control in OSNs. We leverage Facebook as the running example in our discussion because 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 [24].

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]. 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 applica- tions on current OSN platforms can also consume the profile attributes of a user's friends.

Relationship sharing. Another feature of OSNs is that users can share their relationships with other members. Relationships are inherently bidirectional and carry poten- tially 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. Let us consider, for example, a scenario where a user Alice specifies a policy to hide her friend list from the public. However, Bob, one of Alice's friends, specifies a weaker policy that permits his friend list visible to anyone. In this case, if OSNs can solely enforce one party's policy, the relationship between Alice and Bob can still be learned through Bob's friend list. Fig. 1b 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, author- ization requirements from both the owner and the stakeholder should be considered. Otherwise, the stakehol- der's privacy concern may be

violated *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 multi- ple 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 the *owner* of the photo, and Bob and Carol *stakeholders* of the phot



(a) A shared content has multiple stakeholders (b) A shared content is published by a contributor

(c) A disseminator shares other's content published by a contributor.

Fig. 2. MPAC pattern for content sharing

All of them may specify access control policies to control over who can see this photo. Fig. 2a depicts a content sharing pattern where the owner of a content shares the content with other OSN members, and the content has multiple stateholders who may also want to involve in the control of content 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 the *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. Fig. 2b shows a content sharing pattern reflecting this scenario where a contributor publishes a 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.

OSNs also enable users to share others' contents. For example, when Alice views a photo in Bob's space and decides to share this photo with her friends, the photo will be in turn posted in her space and she can specify access control policy 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 compliant with, preventing from the possible leakage of sensitive information via the procedure of data dissemination. Fig. 2c shows a content sharing pattern where the sharing starts with an *originator* (*owner* or *contributor* who uploads the content) publishing the content, and then a disseminator views and shares the content. All access control policies defined by associated users should be enforced to regulate access of the content in dissemina- tor's space. For a more complicated case, the disseminated content may be further *redisseminated* 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 content has been redisseminated, the original access control policies should be always enforced to protect further dissemination of the content.

3 MPAC model for OSNs

In this section, we formalize an MPAC model for OSNs as well as a policy scheme and a policy evaluation mechanism for the specification and enforcement of MPAC policies in OSNs.

3.1 MPAC Model

An OSN can be represented by a relationship network, a set of user groups, and a collection of user data. The relation- ship network of an OSN is a directed labeled graph, where each node denotes a user and each edge represents a relationship between two 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 [27], [26] (or called circles in Google [5]), 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 a web space where users can store and manage their personal data including profile information, friend list and content.

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 $u \in T$.*

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.*

Fig. 3 depicts an example of multiparty social network representation. It describes relationships of five individuals, Alice (*A*), Bob (*B*), Carol (*C*), Dave (*D*), and Edward(*E*), along with their relations with data and their groups of interest. 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 Fig. 3, Alice (*A*) has a direct relationship of type *colleagueOf* with Bob (*B*), whereas Bob (*B*) has a relationship of *friendOf* with Alice (*A*). In addition, two users may have transitive relationship, such as FOF, *colleagues-of-colleagues* and *classmates-of-classmates* (LOL) in this example. Moreover, this example shows that some data items have multiple controllers. For instance, *RelationshipA* has two controllers: the owner, Alice (*A*), and a stakeholder, Carol (*C*). Also, some users may be the controllers of multiple data items. For example, Carol (*C*) is a stakeholder of *RelationshipA* as well as the contributor of *ContentE* . Furthermore, we can notice there are two groups in this example that users can participate in: the "*Fashion*" group and the "*Hiking*" group, and some users, such as Bob (*B*) and Dave (*D*), may join in multiple groups.

3.2. MPAC Policy Specification

To enable a collaborative authorization management of data sharing in OSNs, it is essential for MPAC 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 (RNs) or a set of group names (GNs) in OSNs. We formally define the accessor specification as follows:

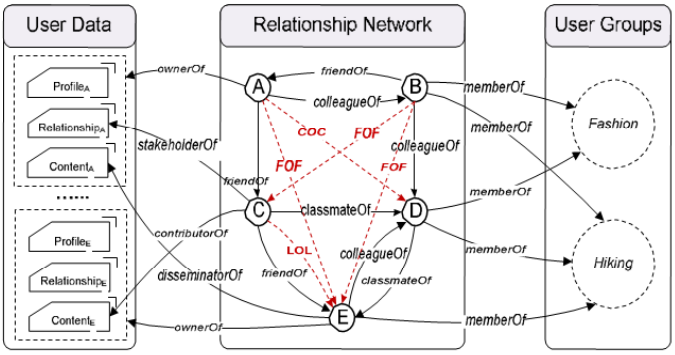


Fig. 3. An example of multiparty social network representation.

Definition 5 (Accessor Specification). *Let ac* 2 *U* [*RT* [*G be a user u* 2 *U, a relationship type rt* 2 *RT, or a group g* 2 *G. Let at* 2 f*UN; RN; GN*g *be the type of*

*the accessor specifica- tion (*user name, relationship type*, and* GN*, respectively). The accessor specification is defined as a set,* accessors ¼ fa1*; . . . ; an*g*, where each element is a tuple <ac;at>.*

Data specification. In OSNs, user data are composed of three types of information: *user profile*, *user relationship*, and *user content*.

To facilitate effective privacy conflict resolution for MPAC, we introduce *sensitivity levels* (*SL*) for data specification, which are assigned by the controllers to the shared data items. A user's judgment of the SL of the data is not binary (private/public), but multidimensional with varying degrees of sensitivity. Formally, the data specification is defined as follows:

3.3 Multiparty Policy Evaluation

Two steps are performed to evaluate an access request over MPAC 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 (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.

The essential reason leading to the conflicts—especially *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. Both of them define their own access control policy stating that 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 when considering multiparty control over the shared data item.

Logical Representation and Analysis of MPAC

In this section, we adopt answer set programming (ASP), a recent form of declarative programming [32], to formally represent our model, which essentially provide a formal foundation of our model in terms of ASP-based representation. Then, we demonstrate how the *correctness* analysis and *authorization* analysis [7] of MPAC can be carried out based on such a logical representation.

Reasoning about MPAC

One of the promising advantages in logic-based representa- tion of access control is that formal reasoning of the authorization properties can be achieved. Once we repre- sent our MPAC model into an ASP program, we can use off- the-shelf ASP solvers to carry out several automated analysis tasks. The problem of verifying an authorization property against our model description can be cast into the problem of checking whether the program has no answer sets, where is the program corresponding to the model specification, *query* is the program corresponding to the program that encodes the negation of the property to check, and *config* is the program that can generate *arbitrary* configurations, for example:

user_attributes(alice,bob,carol,dave, edward).

group_attributes(fashion,hiking). photo_attributes(photoid).

1{user(X):user_attributes(X)}. 1{group(X):group_attributes(X)}. 1{photo(X):photo_attributes(X)}.

Correctness analysis: is used to prove if our proposed access control model is valid.

Example. If we want to check whether the conflict resolution strategy Full consensus permit is correctly defined based on the voting scheme in our model, the input query *query* can be represented as follows:

Decision controllers; permit Þ ¯C CS

If no answer set is found, this implies that the authorization property is verified. Otherwise, an answer set returned by an ASP solver serves as a counterexample that indicates why the description does not entail the authorization property. This helps determine the problems in the model definition.

Authorization analysis: is employed to examine *oversharing* (does current authorization state disclose the data to some users undesirable?) and *undersharing* (does current author- ization state disallow some users to access the data that they are supposed to be allowed?). This analysis service should be incorporated into OSN systems to enable users checking potential authorization impacts derived from collaborative control of shared data.

5 Implementation and Evaluation

5.1 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 cocontrol a shared data item. It is worth noting that our current implementation was restricted to handle

photo sharing in OSNs. Obversely, our approach can be general- ized to deal with other kinds of data sharing, such as videos and comments, in OSNs as long as the stakeholder of shared data is identified with effective methods like tagging or searching. 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 forward 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. *MController* application is based on the iFrame external application approach. Using the Javascript and PHP SDK, it accesses users' Facebook data through the graph API and Facebook query language. Once a user installs *MController* in her/his Facebook space and accepts the necessary permissions, *MController* can access a user's basic informa- tion and contents. Especially, *MController* can retrieve and list all photos, which are owned or uploaded by the user, or where the user was tagged. Once information is imported, the user accesses *MController* through its application page on Facebook, where she/he can query access information, set privacy for photos that she/he is a controller, or view photos she/he is allowed to access.

A core component of *MController* is the decision-making module, which processes access requests and returns responses (either permit or deny) for the requests. Fig. 6 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. 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 Sc 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 corre-

sponding combination strategy discussed previously.

5.2 System Usability and Performance Evaluation

5.2.1 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 ¼ 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.

5.2.2 User Study of M Controller

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 (e.g., "I like the idea of being able to control photos in which I am tagged"). *Simplicity* is a measure how intuitive and useful the system is (e.g., "Setting my privacy settings for a photo in *MController* is Complicated (1) to Simple (5)" with a 5-point scale). *Control* is a measure of the user's perceived control of their personal data (e.g., "If Facebook implemen- ted controls like *MController*'s to control photo privacy, my photos would be better protected"). 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 percent confidence interval for the users' answers. This assumes the population to be mostly normal.

Before using MController. Prior to using *MController*, users were asked a few questions about their usage of Facebook to determine the user's perceived usability of the current Facebook privacy controls. 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 percent positively about the *likability* and *control* of Facebook's privacy management mechanism, and at most 44 percent on Facebook's *simplicity* as shown in Table 1. This demon- strates 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 percent positively about the *likability* and *control*, and at least 67 percent 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*.

5.2.3 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 "FOF" as the accessors instead of "friends." In our experiments, we performed 1,000 inde- pendent 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 *Controller* in Face book.

6 Discussions

In our MPAC 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 SL 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 [12] and Identity Clone attacks have been introduced to OSNs and several

effective approaches have been recently proposed to prevent the former [13], [25] and latter attacks [27], 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 11]) 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 22], [24]. 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.

The need of joint management for data sharing, especially photo sharing, in OSNs has been recognized by the recent work [7], [10], [18], [23]. Squicciarini et al. [36] provided a solution 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. Also, 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. Carminati et al. [8] recently introduced a new class of security policies, called *collaborative security policies*, that basically enhance topology-based access control with respect to a set of collaborative users. In contrast, our work proposes a formal model to address the MPAC 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.

7 Conclusion

In this paper, we have proposed a novel solution for collaborative management of shared data in OSNs.

An MPAC 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. For example, one of our recent work has evaluated the effectiveness of the MPAC conflict resolution approach based on the tradeoff of privacy risk and sharing loss. In addition, users may be involved in the control of a larger number of shared photos and the configurations of the privacy preferences may become time-consuming and tedious tasks. Therefore, we would study inference-based techniques for automatically configure privacy preferences in MPAC. 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/, 2013.

[2] Face book Privacy Policy, http://www.facebook.com/policy. php/, 2013.

[3] Face book Statistics, http://www.facebook.com/press/info. php?statistics, 2013.

[4] Google+ Privacy Policy, http://http://www.google.com/intl/ en/+/policy/, 2013.

[5] The Google+ Project, https://plus.google.com, 2013.

[6] G. Ahn and H. Hu, "Towards Realizing a Formal RBAC Model in Real Systems," *Proc. 12th ACM Symp. Access Control Models and Technologies,* pp. 215-224, 2007.

[7 A. Besmer and H.R. Lipford, "Moving beyond Untagging: Photo Privacy in a Tagged World," *Proc. 28th Int'l Conf. Human Factors in Computing Systems,* pp. 1563-1572, 2010.

[8] B. Carminati and E. Ferrari, "Collaborative Access Control in On-Line Social Networks," *Proc. Seventh Int'l Conf. Collaborative Computing: Networking, Applications and Worksharing (Collaborate- Com),* pp. 231-240, 2011.

[9] B. Carminati, E. Ferrari, and A. Perego, "Rule-Based Access Control for Social Networks," *Proc. Int'l Conf. On the Move to Meaningful Internet Systems,* pp. 1734-1744, 2006.

[10] B. Carminati, E. Ferrari, and A. Perego, "Enforcing Access Control in Web-Based Social Networks," *ACM Trans. Information and System Security,* vol. 13, no. 1, pp. 1-38, 2009.

[11] 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," *IEEE Trans. Multimedia,* vol. 13, no. 1, pp. 14-28, Feb. 2011.

[12] J. Douceur, "The Sybil Attack,"

Proc. Int'l Workshop Peer-to-Peer Systems, pp. 251-260, 2002.

[13] P. Fong, "Preventing Sybil Attacks by Privilege Attenuation: A Design Principle for Social Network Systems," *Proc. IEEE Symp. Security and Privacy (SP),* pp. 263-278, 2011.

[14] P. Fong, "Relationship-Based Access Control: Protection Model and Policy Language," *Proc. First ACM Conf. Data and Application Security and Privacy,* pp. 191-202, 2011.

[15] P. Fong, M. Anwar, and Z. Zhao, "A Privacy Preservation Model for Facebook-Style Social Network Systems," *Proc. 14th European Conf. Research in Computer Security,* pp. 303-320, 2009.

16] M. Harrison, W. Ruzzo, and J. Ullman, "Protection in Operating Systems," *Comm. ACM,* vol. 19, no. 8, pp. 461-471, 1976.

[17] H. Hu and G. Ahn, "Enabling Verification and Conformance Testing for Access Control Model," *Proc. 13th ACM Symp. Access Control Models and Technologies,* pp. 195-204, 2008.

[18] H. Hu and G. Ahn, "Multiparty Authorization Framework for Data Sharing in Online Social Networks," *Proc. 25th Ann. IFIP WG 11.3 Conf. Data and Applications Security and Privacy,* pp. 29-43, 2011.

[19] H. Hu, G.-J. Ahn, and J. Jorgensen, "Enabling Collaborative Data Sharing in Google+.," Technical Report ASU-SCIDSE-12-1, http:// sefcom.asu.edu/mpac/mpac+.pdf, Apr. 2012.

[20] S. Kruk, S. Grzonkowski, A. Gzella, T. Woroniecki, and H. Choi, "D-FOAF: Distributed Identity Management with Access Rights Delegation," *Proc. Asian Semantic Web Conf. (ASWC),* pp. 140-154, 2006.

[21] N. Li, J. Mitchell, and W. Winsborough, "Beyond Proof-of-Compliance: Security Analysis in Trust Management," *J. ACM,* vol. 52, no. 3, pp. 474-514, 2005.

22] B. Qureshi, G. Min, and D. Kouvatsos, "Collusion Detection and Prevention with Fire+ Trust and Reputation Model," *Proc. IEEE 10th Int'l Conf. Computer and Information Technology (CIT),* pp. 2548- 2555, 2010.

23] A. Squicciarini, M. Shehab, and F. Paci, "Collective Privacy Management in Social Networks," *Proc. 18th Int'l Conf. World Wide Web,* pp. 521-530, 2009.

24] E. Staab and T. Engel, "Collusion Detection for Grid Computing," *Proc. Ninth IEEE/ACM Int'l Symp. Cluster Computing and the Grid,* pp. 412-419, 2009.

[25] B. Viswanath, A. Post, K. Gummadi, and A. Mislove, "An Analysis of Social Network-Based Sybil Defenses," *ACM SIGCOMM Computer Comm. Rev.,* vol. 40, pp. 363-374, 2010.

[26] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A Practical Attack to De-Anonymize Social Network Users," *Proc. IEEE Symp. Security and Privacy,* pp. 223-238, 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," *Proc. 18th Int'l Conf. World Wide Web,* pp. 531-540, 2009.