# Detecting and Resolving Privacy Conflicts for Collaborative Data Sharing in Online Social Networks

Hongxin Hu, Gail-Joon Ahn and Jan Jorgensen
Arizona State University
Tempe, AZ 85287, USA
{hxhu,gahn,jan.jorgensen}@asu.edu

## ABSTRACT

We have seen tremendous growth in online social networks (OSNs) in recent years. These OSNs not only offer attractive means for virtual social interactions and information sharing, but also raise a number of security and privacy issues. Although OSNs allow a single user to govern access to her/his data, they currently do not provide any mechanism to enforce privacy concerns over data associated with multiple users, remaining privacy violations largely unresolved and leading to the potential disclosure of information that at least one user intended to keep private. In this paper, we propose an approach to enable collaborative privacy management of shared data in OSNs. In particular, we provide a systematic mechanism to identify and resolve privacy conflicts for collaborative data sharing. Our conflict resolution indicates a tradeoff between privacy protection and data sharing by quantifying privacy risk and sharing loss. We also discuss a proof-of-concept prototype implementation of our approach as part of an application in Facebook and provide system evaluation and usability study of our methodology.

## Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Access controls; H.2.7 [**Information Systems**]: Security, integrity, and protection

## General Terms

Security, Management

## Keywords

Social Networks, Collaborative, Data Sharing, Privacy Conflict, Access Control

## 1. INTRODUCTION

Online social networks (OSNs), such as Facebook, Twitter, and Google+, have become a *de facto* portal for hundreds of millions of Internet users. For example, Facebook, one of representative social network provider, claims that it has more than 800 million active users [3]. With the help of these OSNs, people share personal and public information and make social connections with friends, coworkers, colleagues, family and even with strangers. As a result, OSNs store a huge amount of possibly sensitive and private information on users and their interactions. To protect that information, privacy control has been treated as a central feature of OSNs [2, 4].

OSNs provide built-in mechanisms enabling users to communicate and share information with other members. A typical OSN offers 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 the user 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*, *specific groups* or *everyone* to access their data, relying on their personal privacy requirements.

Despite the fact that OSNs currently provide privacy control mechanisms allowing users to regulate access to information contained in their *own* spaces, users, unfortunately, have no control over data residing *outside* their spaces [7, 15, 21, 22, 24]. 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. Since multiple associated users may have different privacy concerns over the shared data, *privacy conflicts* occur and the lack of collaborative privacy control increases the potential risk in leaking sensitive information by friends to the public.

In this paper, we seek an effective and flexible mechanism to support privacy control of shared data in OSNs. We begin by giving an analysis of data sharing associated with multiple users in OSNs, and articulate several typical scenarios of privacy conflicts for understanding the risks posed by those conflicts. To mitigate such risks caused by privacy conflicts, we develop a collaborative data sharing mechanism to support the specification and enforcement of multiparty privacy concerns, which have not been accommodated by existing access control approaches for OSNs (e.g., [10, 12, 13]). In the meanwhile, a systematic conflict detection and resolution mechanism is addressed to cope with privacy conflicts occurring in collaborative management of data sharing in OSNs. Our conflict resolution approach balances the need for privacy protection and

*ACSAC* '11 Dec. 5-9, 2011, Orlando, Florida USA

the users' desire for information sharing by quantitative analysis of privacy risk and sharing loss. Besides, we implement a proof-of-concept prototype of our approach in the context of Facebook. Our experimental results based on comprehensive system evaluation and usability study demonstrate the feasibility and practicality of our solution.

The rest of the paper is organized as follows. In Section 2, we analyze several conflict scenarios for privacy control in OSNs. In Section 3, we address our proposed mechanism for detecting and resolving privacy conflicts in collaborative data sharing. The details on our prototype implementation and experimental results are described in Section 4. Section 5 gives a brief overview of related work. Section 6 concludes this paper and discusses our future directions.

# 2. PRIVACY CONFLICTS IN ONLINE SOCIAL NETWORKS

Users in OSNs can post statuses and notes, upload photos and videos in their own spaces, tag others to their content, and share the content with their friends. On the other hand, users can also post content in their friends' spaces. The shared content 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 called Alice the *owner* of the photo, and Bob and Carol *stakeholders* of the photo. All of them may be desired to specify privacy policies to control over who can see this photo. 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. Since each associated user may have different privacy concerns over the shared content, privacy conflicts can occur among the multiple users.

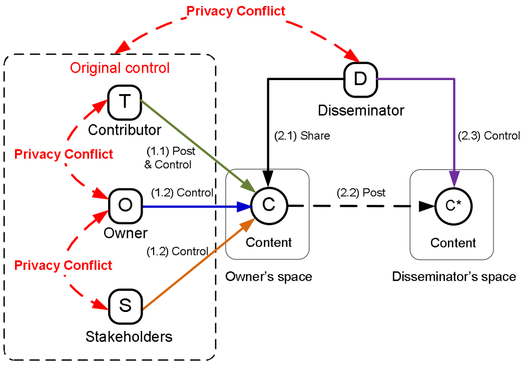


**Figure 1: Privacy Conflicts in OSNs.**

OSNs also enable users to share others' content. 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 to her space and she can 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 privacy concerns of this photo may be violated, resulting in the leakage of sensitive information during the procedure of data dissemination. Figure 1 shows a comprehensive conflict scenario in content sharing where the sharing starts with a *contributor* who uploads the content, and then a disseminator views and shares the content. All privacy conflicts among the *disseminator* and the original controllers (the *owner*, the *contributor* and the *stakeholders*) should be taken into account for regulating access to content in disseminator's space.

In addition to privacy conflicts in *content sharing*, conflicts may also occur in two other situations, *profile sharing* and *friendship sharing*, where multiple parties may have different privacy requirements in sharing their profiles and friendship lists with others or social applications in OSNs.

# 3. OUR APPROACH

Current online social networks, such as Facebook, only allow the data *owner* to fully control the shared data, but lack a mechanism to specify and enforce the privacy concerns from other associated users, leading to privacy conflicts being largely unresolved and sensitive information being potentially disclosed to the public. In this section, we address a collaborative privacy management mechanism for the protection of shared data with respect to multiple controllers in OSNs. A privacy policy scheme is first introduced for the specification and enforcement of multiparty privacy concerns. Then, we articulate our systematic method for identifying and resolving privacy conflicts derived from multiple privacy concerns for collaborative data sharing in OSNs.

## 3.1 Collaborative Control for Data Sharing in OSNs

### 3.1.1 OSN Representation

An OSN can be represented by a friendship network, a set of user groups and a collection of user data. The friendship network of an OSN is a graph, where each node denotes a user and each edge represents a friendship link between two users. Besides, OSNs include an important feature that allows users to be organized in groups [25, 26], 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. We now provide an abstract representation of an OSN with the core components upon which to build our solution:

- $U$ is a set of users of the OSN, $\{u_1, \ldots, u_n\}$. Each user has a unique identifier;
- $G$ is a set of groups to which the users can belong, $\{g_1, \ldots, g_m\}$. Each group also has a unique identifier;
- $UU \subseteq U \times U$ is a binary user-to-user friendship relation;
- $UG \subseteq U \times G$ is a binary user-to-group membership relation;
- $P$ is a collection of user profile sets, $\{p_1, \ldots, p_o\}$, where $p_i = \{p_{i1}, \ldots, p_{ip}\}$ is the profile of a user $i \in U$. Each profile entry is a <*attribute: profile-value*> pair, $p_{ij} =< attr_j : pvalue_j >$, where $attr_j$ is an attribute identifier and $pvalue_j$ is the attribute value;
- $F$ is a collection of user friend sets, $\{f_1, \ldots, f_q\}$, where $f_i = \{u_1, \ldots, u_r\}$ is the friend list of a user $i \in U$;
- $C$ is a collection of user content sets, $\{c_1, \ldots, c_s\}$, where $c_i = \{c_{i1}, \ldots, c_{it}\}$ is a set of content of a user $i \in U$, where $c_{ij}$ is a content identifier; and

- $D$ is a collection of data sets, $\{d_1, \ldots, d_u\}$ , where $d_i = p_i \cup f_i \cup c_i$ is a set of data of a user $i \in U$.

### *3.1.2 Privacy Policy Specification*

To enable a collaborative management of data sharing in OSNs, it is essential for privacy policies to be in place to regulate access over shared data, representing privacy requirements from multiple associated users. Recently, several access control schemes (e.g., [9, 12]) have been proposed to support fine-grained privacy specifications for OSNs. Unfortunately, these schemes can only allow a single user to specify her/his privacy concern. Indeed, a flexible privacy control mechanism in a multi-user environment like OSNs should allow multiple controllers, who are associated with the shared data, to specify privacy policies.

***Controller Specification***: As we discussed previously in the privacy conflict scenarios (Section 2), in addition to the *owner* of data, other controllers, including the *contributor*, *stakeholder* and *disseminator* of data, also need to regulate the access of the shared data. We define these controllers as follows:

DEFINITION 1. *(Owner). Let $e \in d_u$ be a data item in the space of a user $u \in U$ in the social network. The user $u$ is called the owner of e, denoted as $OW_e^u$.*

DEFINITION 2. *(Contributor). Let $e \in d_{u'}$ be a data item published by a user $u \in U$ in the space of another user $u' \in U$ in the social network. The user $u$ is called the contributor of e, denoted as $CB_e^u$.*

DEFINITION 3. *(Stakeholder). Let $e \in d_{u'}$ be a data item in the space of a user $u' \in U$ in the social network. Let $G$ be the set of tagged users associated with e. A user $u \in U$ is called a stakeholder of e, denoted as $ST_e^u$, if $u \in G$.*

DEFINITION 4. *(Disseminator). Let $e \in d_{u'}$ be a data item shared by a user $u \in U$ from the space of another user $u' \in U$ to her/his space in the social network. The user $u$ is called a disseminator of e, denoted as $DS_e^u$.*

Then, we can formally define the controller specification as follows:

DEFINITION 5. *(Controller Specification). Let $cn \in U$ be a user who can regulate the access of data. And let $ct \in CT$ be the type of the cn, where $CT = \{OW, CB, ST, DS\}$ is a set of controller types, indicating Owner, Contributor, Stakeholder and Disseminator, respectively. The controller specification is defined as a tuple $< cn, ct >$.*

***Accessor Specification***: Accessors are a set of users to whom the authorization is granted. Accessors can be represented with a set of user names, the friendship or a set of group names in OSNs. To facilitate collaborative privacy management, we further introduce *trust levels*, which are assigned to accessors when defining the privacy policies. Golbeck [14] discussed how trust could be used in OSNs, focusing on OSNs for collaborative rating. We believe that such considerations can also apply to our privacy management scenario. As addressed in Section 3.2.2, trust is one of the factors in our approach for resolving privacy conflicts. Clearly, in our scenario, trust has a different meaning from the one used in [14]. The notation of trust in our work mainly convey information about how much confidence a controller put on her/his friends who would not disclose the sensitive information to untrusted users. Also, trust levels can be changed in different situations. The notion of accessor specification is formally defined as follows:

DEFINITION 6. *(Accessor Specification). Let ac be a user $u \in U$, the friendship,[1] or a group $g \in G$, that is, $ac \in U \cup \{friendOf\} \cup G$ . Let tl be a trust level, which is a rational number in the range [0,1], assigned to ac. And let $at \in \{UN, FS, GN\}$ be the type of the accessor (*user name*, *friendship*, and* group name*, respectively). The accessor specification is defined as a set, $\{a_1, \ldots, a_n\}$, where each element is a 3-tuple $< ac, tl, at >$.*

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

Again, to facilitate effective resolution of privacy conflicts for collaborative privacy control, we introduce *sensitivity levels* for data specification, which are assigned by the controllers to the shared data. The users' judgment of the sensitivity levels of the data is not binary (private/public), but multi-dimensional with varying degrees of sensitivity. Formally, the data specification is defined as follows:

DEFINITION 7. *(Data Specification). Let $d \in D$ be a data item, and sl be a sensitivity level, which is a rational number in the range [0,1], assigned to d. The data specification is defined as a tuple $< d, sl >$.*

***Privacy Policy***: To summarize the aforementioned features and elements, we introduce a formal definition of privacy policies for collaborative data sharing as follows:

DEFINITION 8. *(Privacy Policy). A privacy policy is a 4-tuple $P =< controller, accessor, data, effect >$, where*

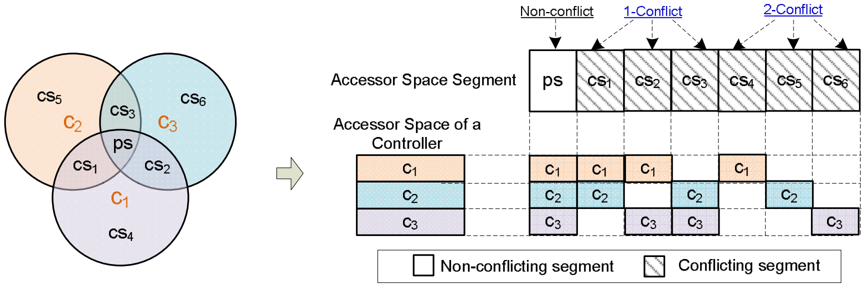
- *$controller$ is a controller specification defined in Definition 5;*
- *$accessor$ is an access specification defined in Definition 6;*
- *$data$ is a data specification defined in Definition 7; and*
- *$effect \in \{permit, deny\}$ is the authorization effect of the policy.*

Suppose the trust levels that a controller can allocate to a user or a user set are {0.00, 0.25, 0.50, 0.75, 1.00}, indicating *none* trust, *weak* trust, *medium* trust, *strong* trust, and *strongest* trust, respectively. Similarly, 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. The following is an example of privacy policy in terms of our policy specification scheme.

EXAMPLE 1. *Alice authorizes users who are her friends or users in hiking group to access a photo (identified by a particular* photoId*) she is tagged in, where Alice considers her friends with a* medium *trust level, the hiking group with a* weak *trust level, and the photo with a* high *sensitivity level:*

$p = (< Alice, ST >, \{< friendOf, 0.5, FS >, < hiking, 0.25, GN >\}, < photoId, 0.75 >, permit)$.

[1] We limit our consideration to $friendOf$ relation. The support of more relations such as $colleagueOf$ and $classmateOf$ does not significantly complicate our approach proposed in this paper.



(a) Accessor space segmentation

(b) Privacy conflict identification

**Figure 2: Example of Privacy Conflict Identification Based on Accessor Space Segmentation.**

## 3.2 Identifying and Resolving 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 associated users 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 a privacy 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 collaborative 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 [24]. Unfortunately, this solution 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 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. Meanwhile, it may reduce the social value of data sharing in OSNs. Therefore, it is important to consider the tradeoff between *privacy protection* and *data sharing* when resolving privacy conflicts. To address this issue, we introduce a mechanism for identifying multiparty privacy conflicts, as well as a systematic solution for resolving multiparty privacy conflicts.

### 3.2.1 Privacy Conflict Identification

Through specifying the privacy policies to reflect the privacy concern, each controller of the shared data item defines a set of trusted users who can access the data item. The set of trusted users represents an *accessor space* for the controller. In this section, we first introduce a space segmentation approach [16] to partition accessor spaces of all controllers of a shared data item into disjoint segments. Then, conflicting accessor space segments (called *conflicting segments* in the rest of this paper), which contain accessors that some controllers of the shared data item do not trust, are identified. Each conflicting segment contains at least one privacy conflict.

Algorithm 1 shows the pseudocode of generating conflicting accessor space segments for all controllers of a shared data item. An entire accessor space derived from the policies of all controllers of shared data item is first partitioned into a set of disjoint segments. As shown in lines 10-28 in Algorithm 1, a function called `Partition()` accomplishes this procedure. This function works by adding an accessor space $s_a$ derived from policies of an controller $a$ to an accessor space set $S$. A pair of accessor spaces must satisfy one of the following relations: *subset* (line 14), *superset* (line 19), *partial match* (line 22), or *disjoint* (line 27). Therefore, one can utilize set operations to separate the overlapped spaces into disjoint spaces.

**Algorithm 1**: Identification of Conflicting Accessor Space

```
   Input: A set of accessor space, $A$.
   Output: A set of disjoined conflicting accessor spaces, $CS$.
1  /* Partition the entire accessor space */
2  $S \longleftarrow Partition(A)$;
3  /* Identify the conflicting segments */
4  $CS.New()$;
5  foreach $s \in S$ do
6      /* Get all controllers associated with a segment s */
7      $C \longleftarrow GetControllers(s)$;
8      if $|C| < |A|$ then
9          $CS.Append(s)$;

10 Partition($A$)
11 foreach $a \in A$ do
12     $s_a \longleftarrow FriendSet(a)$;
13     foreach $s \in S$ do
14         /* $s_a$ is a subset of s*/
15         if $s_a \subset s$ then
16             $S.Append(s \setminus s_a)$;
17             $s \longleftarrow s_a$;
18             $Break$;
19         /* $s_a$ is a superset of s*/
20         else if $s_a \supset s$ then
21             $s_a \longleftarrow s_a \setminus s$;
22         /* $s_a$ partially matches s*/
23         else if $s_a \cap s \neq \emptyset$ then
24             $S.Append(s \setminus s_a)$;
25             $s \longleftarrow s_a \cap s$;
26             $s_a \longleftarrow s_a \setminus s$;
27     $S.Append(s_a)$;
28 return $S$;
```

Conflicting segments are identified as shown in lines 5-9 in Algorithm 1. A set of conflicting segments $CS : \{cs_1, cs_2, \ldots, cs_n\}$ from the policies of conflicting controllers has the following three properties:

1. All conflicting segments are pairwise disjoint: $cs_i \cap cs_j = \emptyset, 1 \leq i \neq j \leq n$;
2. Any two different accessors $a$ and $a'$ within a single conflicting segment ($cs_i$) are defined by the exact same set of controllers: $GetController(a) = GetController(a')$, where

$a \in cs_i, a^{'} \in cs_i, a \neq a^{'}$; [2] and

3. The accessors in any conflicting segments are untrusted by at least one controller of the shared data item.

Figure 2 gives an example of identifying privacy conflicts based on accessor space segmentation. We use circles to represent accessor spaces of three controllers, $c_1$, $c_2$ and $c_3$, of a shared data item. We can notice that three of accessor spaces overlap with each other, indicating that some accessors within the overlapping spaces are trusted by multiple controllers. After performing the space segmentation, seven disjoint accessor space segments are generated as shown in Figure 2 (a). To represent privacy conflicts in an intuitive way, we additionally introduce a grid representation of privacy conflicts, in which space segments are displayed along the horizontal axis of a matrix, controllers are shown along the vertical axis of the matrix, and the intersection of a segment and a controller is a grid that displays the accessor subspace covered by the segment. We classify the accessor space segments as two categories: *non-conflicting* segment and *conflicting* segment. *Non-conflicting* segment covers all controllers' access spaces, which means any accessor within the segment is trusted by all controllers of the shared data item, indicating no privacy conflict occurs. A *conflicting* segment does not contain all controllers' access spaces that means accessors in the segment are untrusted by some controllers. Each *untrusting* controller points out a privacy conflict. Figure 2 (b) shows a grid representation of privacy conflicts for the example. We can easily identify that the segment *ps* is a *non-conflicting* segment, and $cs_1$ through $cs_6$ are *conflicting* segments, where $cs_1$, $cs_2$ and $cs_3$ indicate *one* privacy conflict, respectively, and $cs_4$, $cs_5$ and $cs_6$ are associated with *two* privacy conflicts, respectively.

### 3.2.2 Privacy Conflict Resolution

The process of privacy conflict resolution makes a decision to allow or deny the accessors within the conflicting segments to access the shared data item. In general, allowing the assessors contained in conflicting segments to access the data item may cause *privacy risk*, but denying a set of accessors in conflicting segments to access the data item may result in *sharing loss*. Our privacy conflict resolution approach attempts to find an optimal tradeoff between privacy protection and data sharing.

***Measuring Privacy Risk***: The privacy risk of a conflicting segment is an indicator of potential threat to the privacy of controllers in terms of the shared data item: the higher the privacy risk of a conflicting segment, the higher the threat to controllers' privacy. Our basic premises for the measurement of privacy risk for a conflicting segment are the following: (a) the lower the number of controllers who trust the accessors within the conflicting segment, the higher the privacy risk; (b) the stronger the general privacy concerns of controllers, the higher the privacy risk; (c) the more sensitive the shared data item, the higher the privacy risk; (d) the wider the data item spreads, the higher the privacy risk; and (e) the lower the trust levels of accessors in the conflicting segment, the higher the privacy risk. Therefore, the privacy risk of a conflicting segment is calculated by a monotonically increasing function with the following parameters:

- **Number of privacy conflicts**: The number of privacy conflicts in a conflicting segment is indicated by the number of the untrusting controllers. The untrusting controllers of a conflict segment $i$ are returned by a function $controllers_{ut}(i)$;
- **General privacy concern of an untrusting controller**: The general privacy concern of an untrusting controller $j$ is denoted as $pc_j$. The general privacy concern of a controller can be derived from her/his *default* privacy setting for data sharing. Different controllers may have different general privacy concern with respect to the same kinds of data. For example, public figures may have higher privacy concern on their shared photos than ordinary people;
- **Sensitivity of the data item**: Data sensitivity in a way defines controllers' perceptions of the confidentiality of the data being transmitted. The sensitivity level of the shared data item explicitly chosen by an untrusting controller $j$ is denoted as $sl_j$. The factor depends on the untrusting controllers themself. Some untrusting controllers may consider the shared data item with the higher sensitivity;
- **Visibility of the data item**: The visibility of the data item with respect to a conflicting segment captures how many accessors are contained in the segment. The more the accessors in the segment, the higher the visibility; and
- **Trust of an accessor**: The trust level of an accessor $k$ is denoted as $tl_k$, which is an average value of the trust levels defined by the trusting controllers of the conflicting segment for the accessor.

The privacy risk of a conflict segment $i$ due to an untrusting controller $j$, denoted as $PR(i, j)$, is defined as

$$PR(i,j) = pc_j \otimes sl_j \otimes \sum_{k \in accessors(i)} (1 - tl_k) \quad (1)$$

where, function $accessors(i)$ returns all accessors in a segment $i$, and operator $\otimes$ is used to represent any arbitrary combination functions. For simplicity, we utilize the product operator.

In order to measure the *overall* privacy risk of a conflicting segment $i$, denoted as PR(i), we can use following equation to aggregate the privacy risks of $i$ due to different untrusting controllers. Note that we can also use any combination function to combine the per-controller privacy risk. For simplicity, we employ the summation operator here.

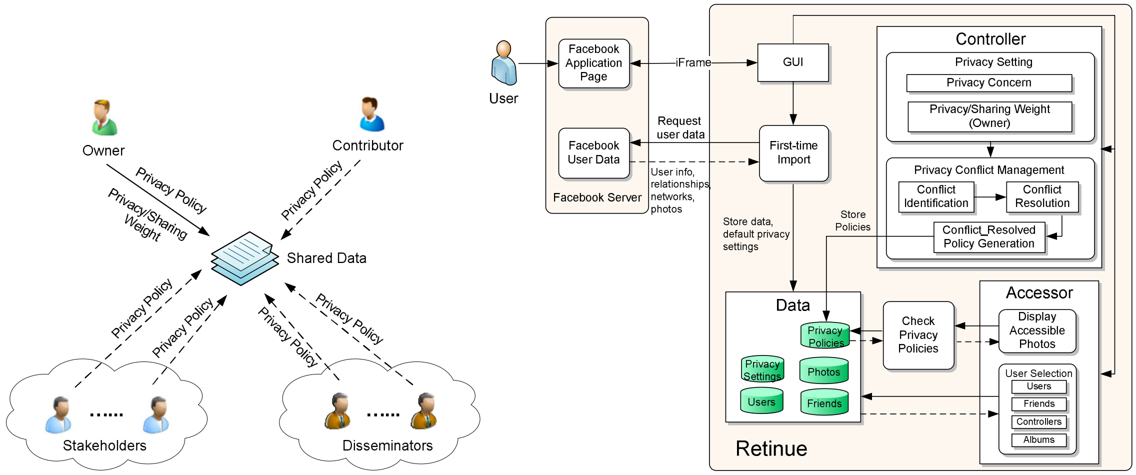
$$\begin{aligned} PR(i) &= \sum_{j \in controllers_{ut}(i)} (PR(i,j)) \\ &= \sum_{j \in controllers_{ut}(i)} (pc_j \times sl_j \times \sum_{k \in accessors(i)} (1 - tl_k)) \end{aligned} \quad (2)$$

***Measuring Sharing Loss***: When the decision of privacy conflict resolution for a conflicting segment is "deny", it may cause losses in potential data sharing, since there are controllers expecting to allow the accessors in the conflicting segment to access the data item. Similar to the measurement of the privacy risk, five factors are adopted to measure the sharing loss for a conflicting segment. Compared with the factors used for quantifying the privacy risk, the only difference is that we will utilize a factor, *number of trusting controllers*, to replace the factor, *number of privacy conflicts (untrusting controllers)*, for evaluating the sharing loss of a conflicting segment. The overall sharing loss $SL(i)$ of a conflicting segment $i$ is computed as follows:

$$SL(i) = \sum_{j \in controllers_t(i)} ((1 - pc_j \times sl_j) \times \sum_{k \in accessors(i)} tl_k) \quad (3)$$

where, function $controllers_t(i)$ returns all trusting controllers of a segment $i$.

[2] $GetController()$ is a function that returns all controllers whose accessor spaces contain a specific accessor.



(a) Collaborative Control Overview

(b) Operational Components in *Retinue* Application

**Figure 3: System Architecture of *Retinue*.**

***Privacy Conflict Resolution on the Tradeoff between Privacy Protection and Data Sharing***: The tradeoff between privacy and utility in data publishing has been recently studied [8, 19]. Inspired by those work, we introduce a mechanism to balance privacy protection and data sharing for an effective privacy conflict resolution in OSNs.

An optimal solution for privacy conflict resolution should cause a little more privacy risk when allowing the accessors in some conflicting segments to access the data item, and gets lesser loss in data sharing when denying the accessors to access the shared data item. Thus, for each conflict resolution solution $s$, a resolving score $RS(s)$ can be calculated using the following equation:

$$RS(s) = \frac{1}{\alpha \sum_{i_1 \in CS_p^s} PR(i_1) + \beta \sum_{i_2 \in CS_d^s} SL(i_2)} \quad (4)$$

where, $CS_p^s$ and $CS_d^s$ denote *permitted* conflicting segments and *denied* conflicting segments respectively in the conflict resolution solution $s$. And $\alpha$ and $\beta$ are preference weights for the privacy risk and the sharing loss, $0 \leq \alpha, \beta \leq 1$ and $\alpha + \beta = 1$.

Then, the optimal conflict resolution $CR_{opt}$ on the tradeoff between privacy risk and sharing loss can be identified by finding the maximum resolving score:

$$CR_{opt} = \max_{s} RS(s) \quad (5)$$

To find the maximum resolving score, we can first calculate the privacy risk ($PR(i)$) and the sharing loss ($SL(i)$) for each conflict segment ($i$), individually. Finally, following equation can be utilized to make the decisions (permitting or denying conflicting segments) for privacy conflict resolution, guaranteeing to always find an optimal solution.

$$Decision = \begin{cases} \text{Permit} & \text{if } \alpha SL(i) \geq \beta PR(i) \\ \text{Deny} & \text{if } \alpha SL(i) < \beta PR(i) \end{cases} \quad (6)$$

### 3.2.3 *Generating Conflict-Resolved Policy*

Once the privacy conflicts are resolved, we can aggregate accessors in *permitted* conflicting segments $CS_p$ and accessors in the *non-conflicting* segment $ps$ (in which accessors should be always allowed to access the shared data item) together to generate a new accessor list ($AL$) as follows:

$$AL = (\bigcup_{i \in CS_p} Accessors(i)) \cup Accessors(ps) \quad (7)$$

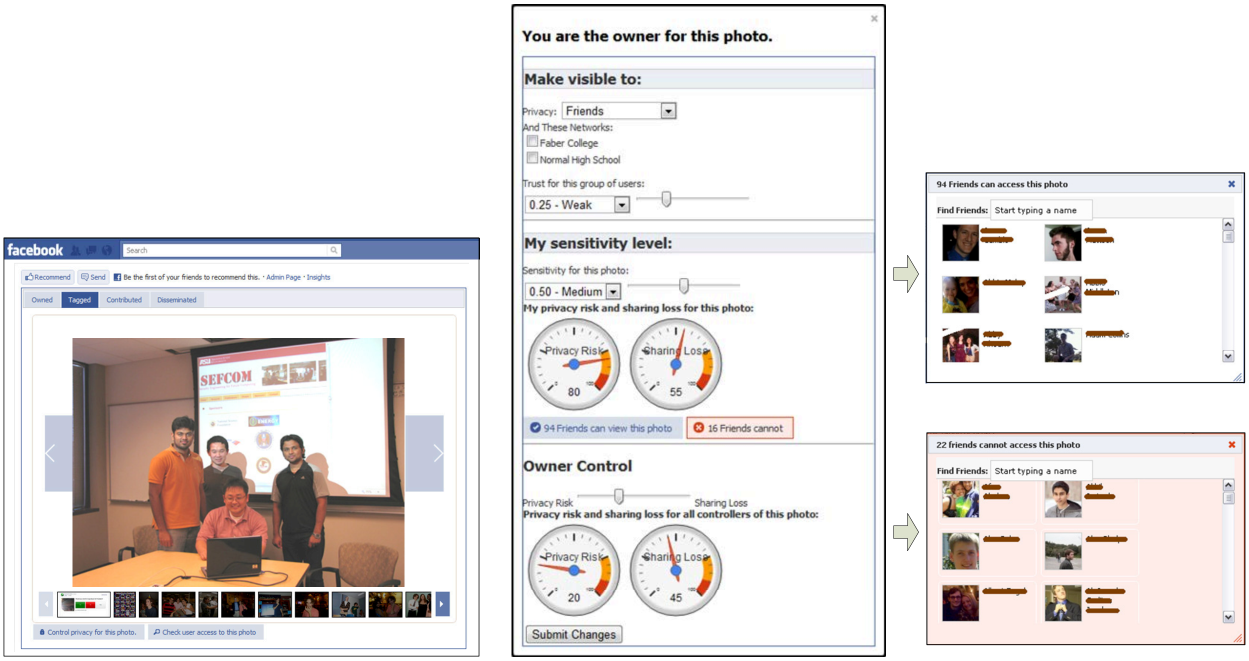
Using the example shown in Figure 2, we assume that $cs_1$ and $cs_3$ become *permitted* conflicting segments after resolving the privacy conflicts. Therefore, the aggregated accessor list can be derived as $AL = Accessors(cs_1) \cup Accessors(cs_3) \cup Accessors(ps)$. Finally, the aggregated accessor list is used to construct a conflict-resolved privacy policy for the shared data item. The generated policy will be leveraged to evaluate all access requests toward the data item.

## 4. IMPLEMENTATION AND EVALUATION

### 4.1 Prototype Implementation

We implemented a proof-of-concept Facebook application for the collaborative management of shared data called *Retinue* (http://apps.facebook.com/retinue_tool). Our prototype application enables multiple associated users to specify their privacy concerns to co-control a shared data item. *Retinue* is designed as a third-party Facebook application which is hosted in an Apache Tomcat application server supporting PHP and MySQL databases, with a user interface built using jQuery and jQuery UI and built on an AJAX-based interaction model. *Retinue* 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. It is worth noting that our current implementation was restricted to handle photo sharing in OSNs. Obversely, our approach can be generalized to deal with other kinds of data sharing (e.g. videos and comments) in OSNs as long as the stakeholder of shared data are identified with effective methods like tagging or searching.

Figure 3 shows the system architecture of *Retinue*. The overview of collaborative control process is depicted in Figure 3(a), where the *owner* can regulate the access of the shared data. In addition, other controllers, such as *the contributor*, *stakeholders* and *disseminators*, can specify their privacy concerns over the shared data as well. To effectively resolve privacy conflicts caused by different privacy concerns of multiple controllers, the data *owner* can also adjust the preference weights for the privacy risk and the sharing


facebook
Search
Recommend
Send
Be the first of your friends to recommend this. · Admin Page · Insights
Owned
Tagged
Contributed
Disseminated
SEFCOM
Control privacy for this photo.
Check user access to this photo

(a) Main Interface.

You are the owner for this photo.

Make visible to:

Privacy: Friends

And These Networks:

□ Faber College

□ Normal High School

Trust for this group of users: 0.25 - Weak

My sensitivity level:

Sensitivity for this photo: 0.50 - Medium

My privacy risk and sharing loss for this photo:

Privacy Risk 80

Sharing Loss 55

94 Friends can view this photo

16 Friends cannot

Owner Control

Privacy Risk — Sharing Loss

Privacy risk and sharing loss for all controllers of this photo:

Privacy Risk 20

Sharing Loss 45

Submit Changes

94 Friends can access this photo

Find Friends: Start typing a name

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

22 friends cannot access this photo

Find Friends: Start typing a name

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

(b) Controllers' Interfaces.

**Figure 4: *Retinue* Interfaces.**

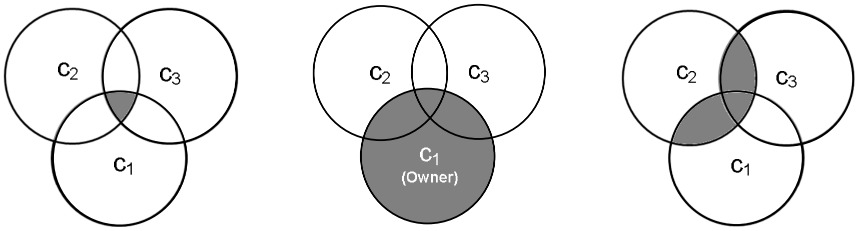
loss to make an appropriate privacy-sharing tradeoff. Figure 3(b) shows the core components in *Retinue* application and their interactions. The *Retinue* application is hosted on an external website, but is accessed on a Facebook application frame via an iFrame. The Facebook server handles login and authentication for the application, and other user data is imported on the user's first login. At this point, users are asked to specify their initial privacy settings and concerns for each type of photo. All photos are then imported and saved using these initial privacy settings. Users' networks and friend lists are imported from Facebook server as well. Once information is imported, a user accesses *Retinue* through the application page on Facebook, where s/he can query access information, complete privacy setting for photos in which s/he is a controller, and view photos s/he is allowed to access. The component for privacy conflict management in *Retinue* application is responsible for the privacy conflict detection and resolution, and the generation of conflict-resolved privacy policy, which is then used to evaluate access requests for the shared data.

A snapshot of the main interface of *Retinue* is shown in Figure 4 (a). All photos are loaded into a gallery-style interface. To access photos, a user clicks the "Access" tab and then s/he can view her/his friends' photos that s/he was authorized. To control photo sharing, a user clicks the "Owned", "Tagged", "Contributed", or "Disseminated" tabs, then selects any photo in the gallery to define her/his privacy preferences for that photo. The controllers' interfaces are depicted in Figure 4 (b). A controller can select the trusted groups of accessors and assign corresponding trust levels, as well as choose the sensitivity level for the photo. Also, the privacy risk and sharing loss for the controller with respect with the photo are displayed in the interface. In addition, the controller can immediately see how many friends can or cannot access the photo in the interface. If the controller clicks the buttons, which show the numbers of accessible or unaccessible friends, a window appears showing the details of all friends who can or cannot view the photo. The purpose of such feedback information is not only to give a controller the information of how many friends can or cannot access the photo, but as a way to react to results. If the controller is not satisfied with the current situation of privacy control, s/he may adjust her/his privacy settings, contact the owner of the photo to ask her/him to change the weights for the privacy risk and the sharing loss, or even report a privacy violation to request OSN administrators to delete the photo. If the user is the owner of the photo, s/he can also view the overall privacy risk and sharing loss for the shared photo, and has the ability to adjust the weights to balance privacy protection and data sharing of the shared photo.

## 4.2 Evaluation and Experiments

### 4.2.1 Evaluation of Privacy Conflict Resolution

We evaluate our approach for privacy conflict resolution by comparing our solution with the *naive solution* and the privacy control solution used by existing OSNs, such as Facebook (simply called *Facebook solution* in the rest of this paper) with respect to two metrics, *privacy risk* and *sharing loss*. Consider the example demonstrated in Figure 2, where three controllers desire to regulate access of a shared data item. The *naive solution* is that only the accessors in the non-conflicting segment are allowed to access the data item as shown in Figure 5(a). Thus, the *privacy risk* is always equal to 0 for this solution. However, the *sharing loss* is the absolute maximum, as all conflicting segments, which may be allowed by at least one controller, are always denied. The *Facebook solution* is that the owner's decision has the highest priority. All accessors within the segments covered by the owner's space are allowed to access the data item, but all other accessors are denied as illustrated in Figure 5(b). This is, obviously, ideal for the owner, since her/his *privacy risk* and *sharing loss* are both equal to 0. However, the *privacy risk* and the *sharing loss* are large for every non-owner controller.

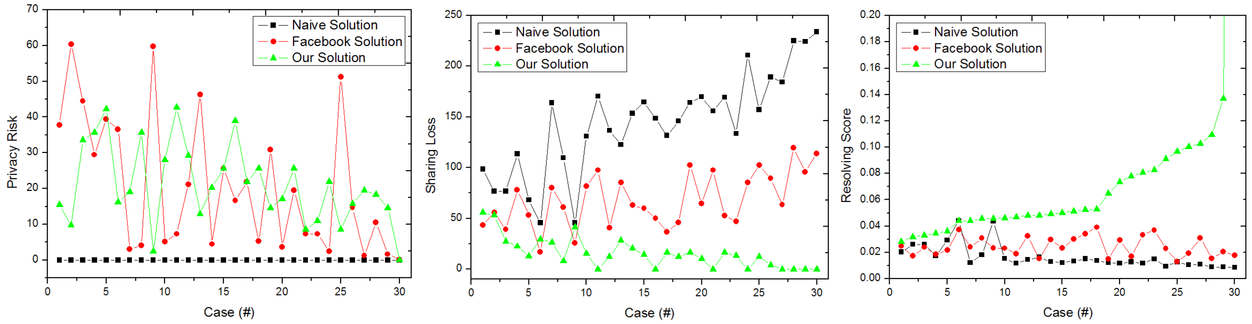


(a) Naive Solution.

(b) Facebook Solution.

(c) Our Solution.

**Figure 5: Example of Resolving Privacy Conflicts.**



(a) Privacy Risk.

(b) Sharing Loss.

(c) Resolving Score.

**Figure 6: Conflict Resolution Evaluation.**

For our solution, each conflicting segment is evaluated individually. Using the same example given in Figure 2, suppose $cs_1$ and $cs_3$ become *permitted* conflicting segments after resolving the privacy conflicts. Figure 5(c) demonstrates the result of our privacy conflict resolution. Our solution make a tradeoff between privacy protection and data sharing by maximizing the resolving score, which is a combination of *privacy risk* and *sharing loss*. The worst case of our solution is the same as the *naive solution*–only mutually permitted accessors are allowed to access the data item. However, this case only occurs when strong privacy concerns are indicated by each controller. On the other hand, if all accessors have pretty weak privacy concerns, all accessors in conflicting segments may be allowed to access the data, which is not possible with either of other two solutions. Such a case leads to a sharing loss of 0, but does not have an significantly increased privacy risk against other two solutions.

To quantitatively evaluate our solution, our experiment used cases where there are three controllers of shared data items and assume that each controller has indicated to allow her/his friends to view the data item. We also utilized the average number of user friends, 130, which is claimed by Facebook statistics [3]. Additionally, we assume all controllers share 30 friends with each other, 10 of which are shared among everyone (common users). All settings including privacy concerns, sensitivity levels, and trust levels were randomized for each case, and the privacy risk, sharing loss, and resolving score for each case were calculated. To represent the data sensibly, we sorted the samples from lowest resolving score to highest under our evaluation. Figure 6 shows our experimental results with respect to randomly generated 30 user cases.

In Figure 6(a), we can observe that the privacy risks for the *naive solution* are always equal to 0, since no untrusted accessors are allowed to view the data item. The privacy risks for *Facebook solution* and our solution wavered. Obviously, this depends greatly on the settings of the non-owner controllers. If these controllers are apathetic toward the shared data item, *Facebook solution* will be preferable. However, it should be noted that *Facebook solution* had very high extrema. This is avoided in our solution where high privacy risks will usually result in denying access.

Unsurprisingly, the sharing loss for the *naive solution* was always the highest, and often higher than both other two solutions as shown in Figure 6(b). Our solution usually had the lowest sharing loss, and sometimes is equivalent to the *naive* or *Facebook solution*, but rarely greater than. One may notice that the sharing loss is very low compared to the privacy risks in our experience. This is an inherent effect of our solution itself–if sharing loss is very high, users will be granted access to the data item, changing this segment's sharing loss to zero.

As we can notice from Figure 6(c), the resolving score for our solution is always as good as or better than the *naive* or *Facebook solution*. In our sample data, it was usually significantly better, and rarely was the same as either of other two solutions. It further indicates that our solution can always achieve a good tradeoff between privacy protection and data sharing for privacy conflict resolution.

### 4.2.2 Evaluation of System Usability

***Participants and Procedure***: *Retinue* 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=30) to explore the factors surrounding users' desires for privacy controls such as those implemented in *Retinue*. Particularly, 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 ran-

**Table 1: Usability Evaluation for Facebook and *Retinue* Privacy Controls.**

| Metric | | Facebook | | Retinue | |
|---|---|---|---|---|---|
| | | Average | Upper bound on 95% confidence interval | Average | Lower bound on 95% confidence interval |
| Likability | | 0.39 | 0.44 | 0.74 | 0.72 |
| Understanding | | 0.33 | 0.36 | 0.69 | 0.65 |
| Control | Sharing with Trusted Users | 0.36 | 0.40 | 0.69 | 0.66 |
| | Protecting from Untrusted Users | 0.30 | 0.35 | 0.71 | 0.70 |

dom sampling, recruiting using the natural dissemination features of Facebook arguably gives an accurate profile of the ecosystem.

In our user study (http://bit.ly/retinue_study), participants were asked to first answer some questions about their usage and perception of Facebook's privacy controls. 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, set privacy settings for a photo they own, and answer questions about their understanding. As users completed these actions, they were asked questions on the usability of the controls in Retinue.

***User Study of Retinue***: The criteria for usability evaluation were split into three areas: *likeability*, *understanding*, and *control*. *Likeability* is simply a measure of a user's basic opinion of a particular feature or control. While this does not provide specific feedback for improvement, it can help identify what aspects of sharing and control are important to a user. *Understanding* is a measure of how intuitive the concepts and controls are. This is tremendously useful for improving the usability of controls. *Control* is a measure of the user's perceived control of their personal data. *Control*, in addition, can be narrowed down into the areas of *sharing with trusted users* and *protecting from untrusted users*. While this is not a definitive measure of privacy, making a user feel safe is almost as important as protecting a user. Questions were measured on a three- or four-point scale (scaled from 0 to 1 for numerical analysis). For measurement analysis, a higher number is used to indicate a positive opinion or perception, while a lower number is used to indicate a negative one. We were interested in the average user perception of the system, so we analyzed a 95% confidence interval for the users' answers. This assumes the population to be mostly normal.

Before using *Retinue*, users were asked a few questions about their usage of Facebook to determine the user's perceived usability of the current Facebook's privacy controls. This included questions on *likeability* (e.g. "indicate how much you like privacy features for photos you are tagged in"), *understanding* (e.g. "indicate how much you understand how to prevent certain people from seeing photos I am tagged in"), and *control* (e.g. "indicate how in control you feel when sharing photos I own with people I want to"). For our confidence interval, we were interested in determining the average user's maximum positive opinion of Facebook's privacy controls, so we looked at the upper bound of the confidence interval.

An average user asserts at most 44% positively about the *likability*, 40% positively about *sharing control*, 35% positively about *protection control* and 36% on their *understanding* of Facebook's privacy mechanisms as shown in Table 1. This demonstrates an average negative opinion of the Facebook's privacy controls that users currently must use.

After Using *Retinue*, users were then asked to perform a few tasks in *Retinue* and were asked a few questions to determine the users perceived usability of *Retinue*. This also included questions on *likeability* (e.g. "indicate how much you like the *trust level* feature"), *understanding* (e.g. "indicate your understanding of the meaning of *sharing loss*"), and *control* (e.g. "please indicate how in control you feel when sharing photos I own with the people I want to"). For our confidence interval, we were interested in determining the average user's minimum positive opinion of *Retinue*'s privacy controls, so we looked at the lower bound of the confidence interval.

An average user asserts at least 72% positively on *likeability*, 65% positively on *understanding*, 66% on *sharing control* and 70% on *protection control* as shown in Table 1. This demonstrates an average positive opinion of the controls and ideas presented to users in *Retinue*.

## 5. RELATED WORK

Several proposals of an access control scheme for OSNs have been introduced (e.g., [9, 10, 12, 13, 17]). Carminati et al. [9] introduced a trust-based access control mechanism, which 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 system and a related enforcement mechanism for controlled sharing of information in OSNs [10]. Fong et al. [13] proposed an access control model that formalizes and generalizes the access control mechanism implemented in Facebook. Gates [11] described relationship-based access control as one of the new security paradigms that addresses the requirements of the Web 2.0. Then, Fong [12] recently formulated this paradigm called a Relationship-Based Access Control (ReBAC) that bases authorization decisions on the relationships between the resource owner and the resource accessor in an OSN. However, none of these work could accommodate privacy control requirements with respect to the *collaborative* data sharing in OSNs.

Several recent work [7, 15, 18, 22, 24] recognized the need of joint management for data sharing, especially photo sharing, in OSNs. In particular, Squicciarini et al. [22] proposed a solution for collective privacy management for photo sharing in OSNs. This work considered the privacy control 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 for shared content. Game theory was applied to evaluate the scheme. However, a general drawback of this 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 only the winning bids could determine who was able to access the data, instead of accommodating all stakeholders' privacy preferences. In contrast, our work proposes a simple but flexible mechanism for collaborative management of shared data in OSNs. In particular, we introduce an effective conflict resolution solution, which makes a tradeoff between privacy protection and data sharing considering the privacy concerns from multiple associated users.

Measuring privacy risk in OSNs has been addressed recently by several work [6, 20, 23]. Becker et al. [6] presented *PrivAware*, a tool to detect and report unintended information loss through quantifying privacy risk associated with friend relationship in OSNs. In [23], Talukder et al. discussed a privacy protection tool, called *Privometer*, which can measure the risk of potential privacy leakage cased by malicious applications installed in the user's friend profiles and suggest self-sanitization actions to lessen this leakage accordingly. Liu et al. [20] proposed a framework to compute the privacy score of a user, indicating the user's potential risk caused by her/his participation in OSNs. Their solution also focused on the privacy settings of users with respect to their profile items. Compared with those existing work, our approach measures the privacy risk caused by different privacy concerns from multiple users, covering profile sharing, friendship sharing, as well as content sharing in OSNs.

# 6. CONCLUSION

In this paper, we have proposed a novel solution for privacy conflict detection and resolution for collaborative data sharing in OSNs. Our conflict resolution mechanism considers privacy-sharing tradeoff by quantifying privacy risk and sharing loss. Also, we have described a proof-of-concept implementation of our solution called *Retinue*, along with the extensive evaluation of our approach. As part of future work, we will formulate a comprehensive access control model to capture the essence of collaborative authorization requirements for data sharing in OSNs. Also, we would extend our work to address security and privacy challenges for emerging information sharing services such as location sharing [1] and other social network platforms such as Google+ [5].

## Acknowledgments

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).

# 7. REFERENCES

[1] Facebook Places. `http://www.facebook.com/places/`.

[2] Facebook Privacy Policy. `http://www.facebook.com/policy.php/`.

[3] Facebook Statistics. `http://http://www.facebook.com/press/info.php?statistics`.

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

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

[6] J. Becker and H. Chen. Measuring privacy risk in online social networks. In *Proceedings of the 2009 Workshop on Web*, volume 2. Citeseer.

[7] 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.

[8] J. Brickell and V. Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *Proceeding of the 14th ACM SIGKDD*, pages 70–78. ACM, 2008.

[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] 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.

[13] 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.

[14] 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.

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

[16] H. Hu, G. Ahn, and K. Kulkarni. Anomaly discovery and resolution in web access control policies. In *Proceedings of the 16th ACM symposium on Access control models and technologies*, pages 165–174. ACM, 2011.

[17] 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.

[18] A. Lampinen, V. Lehtinen, A. Lehmuskallio, and S. Tamminen. We're in it together: interpersonal management of disclosure in social network services. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 3217–3226. ACM, 2011.

[19] T. Li and N. Li. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD*, pages 517–526. ACM, 2009.

[20] K. Liu and E. Terzi. A framework for computing the privacy scores of users in online social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(1):6, 2010.

[21] M. Madejski, M. Johnson, and S. Bellovin. The Failure of Online Social Network Privacy Settings. Technical Report CUCS-010-11, Columbia University, NY, USA. 2011.

[22] 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.

[23] N. Talukder, M. Ouzzani, A. Elmagarmid, H. Elmeleegy, and M. Yakout. Privometer: Privacy protection in social networks. In *Proceedings of 26th International Conference on Data Engineering Workshops (ICDEW)*, pages 266–269. IEEE, 2010.

[24] K. Thomas, C. Grier, and D. Nicol. unFriendly: Multi-party Privacy Risks in Social Networks. In *Privacy Enhancing Technologies*, pages 236–252. Springer, 2010.

[25] 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.

[26] 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.