Enabling Collaborative Data Sharing in Google+ (Technical Report, SEFCOM, March 2012)

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

ABSTRACT

Most of existing online social networks, such as Facebook and Twitter, are designed to bias towards information disclosure to a large audience. Google recently launched a new social network platform, Google+. By introducing the notion of 'circles', Google+ enables users to selectively share data with specific groups within their personal network, rather than sharing with all of their social connections at once. Although Google+ can help mitigate the gap between the individuals' expectations and their actual privacy settings, it still only allows a single user to restrict access to her/his data but cannot provide any mechanism to enforce privacy concerns over data associated with multiple users. In this paper, we propose an approach to facilitate collaborative privacy management of shared data in Google+. We formulate a circle-based multiparty access control model (CMAC) to capture the essence of collaborative authorization requirements in Google+, along with a multiparty policy specification scheme and a policy enforcement mechanism. We also discuss a proof-of-concept prototype of our approach and describe system evaluation and usability study of our prototype.

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

Google+, Social Networks, Access Control, Collaborative, Privacy

1. INTRODUCTION

Online social networks (OSNs) have experienced explosive growth in recent years and become a *de facto* portal for hundreds of millions of Internet users. Facebook, for example, claims that it has more than 800 million active users [2]. As the popularity of OSNs continues to grow, a huge amount of possibly sensitive and private information has been uploaded to OSNs. To protect such a large volume of sensitive information, access control has received considerable attention as a central feature of OSNs [1, 4].

Today, nearly 4 out of 5 active Internet users visit OSNs [11], leading to a fundamental shift in the patterns of information exchange over the Internet. Users in OSNs are now required to be content *creators* and *managers*, rather than just being content *consumers*. A typical OSN allows users to create connections to 'friends', thereby sharing with them a wide variety of personal information. These connections, however, rarely distinguish between different types of relationship. Even within a network of 'friends', users may want to regulate the sharing of information with different people based on their different relationships. Unfortunately, most of exiting OSNs could not provide effective mechanisms to sufficiently address how to organize people and how to utilize relationships for privacy settings. For example, Facebook has introduced an *optional* feature called *Friend Lists* which allows us to group friends and specify whether a piece of data should be visible or invisible to a particular friend list [6]. However, studies have consistently shown that users struggle to adopt this feature for managing their friends and customizing their privacy settings [12, 23, 39], due in part to an unintuitive and convoluted process [33, 42]. As a result, significant privacy violations and mismatched user expectations in OSNs were identified [35, 36, 45]. In particular, as demonstrated in [35], less than 40% of privacy settings match users' expectations in Facebook.

To address the above-mentioned issue, Google recently launched a new social network service, namely Google+ [5], by utilizing 'circles' as its fundamental design feature for sorting connections and enabling users to selectively share the information with their friends, family, colleagues, etc, instead of sharing with all of their connections. Circles even demonstrate how users can accommodate different roles that they normally play in their daily life. Especially, compared with Facebook's user management, the intuitive user interface of Google+ helps a user manage his/her OSN friends through an easy drag-and-drop function.

Despite the fact that Google+ can help mitigate the gap between the users' expectations and their actual privacy settings, it still only allows a single user to regulate access to information contained in their *own* spaces but cannot provide control over data residing *outside* their spaces. For instance, if a user posts a comment in a friend's space, s/he cannot specify who can view the comment. Furthermore, when a user uploads a photo and tags friends who appear in the photo, the tagged friends cannot govern who can see this photo, even though the tagged friends may have different privacy concerns about the photo. To address such a critical issue, Google+ has offered several preliminary protection mechanisms. For example, Google+ allows tagged users to remove the tags linked to their profiles or report violations asking Google+ managers to remove the contents that they do not want to share with the public. However, these simple protection mechanisms suffer from several limitations. On one hand, removing a tag from a photo can only prevent other members from viewing a user's profile by means of the association link, but the user's image is still contained in the photo. Since original access control policies cannot be changed, the user's image continues to be disclosed to all authorized users. On the other hand, reporting to Google+ only allows to either keep or delete the content. Such a binary decision from Google+ managers is either too loose or too restrictive, relying on the Google+'s administration and requiring several people to report their request on the same content. In another example, the first privacy flaw in

Google+ was identified in [8] and this flaw implies that any content shared with a particular circle could be reshared with *anyone* by someone from those circles. This problem was fixed by Google+ by disabling limited content to be sharable publicly. However, this solution still cannot prevent users who can access the shared content from disseminating the content to anyone in their circles, which may violate the original content owner's privacy control. Hence, it is essential to develop an effective and flexible access control mechanism for Google+, accommodating the special authorization requirements coming from multiple associated users for managing the shared data collaboratively.

In this paper, we attempt to explore a systematic method to enable collaborative management of shared data in Google+. A circle-based multiparty access control (CMAC) model is formulated for Google+ to capture the core features of multiparty authorization requirements which have not been accommodated in most of existing access control systems for OSNs so far (e.g., [19, 20, 24, 25]). Our model also contains a multiparty policy specification scheme and a policy evaluation mechanism. Since policy conflicts are inevitable in multiparty authorization enforcement, a conflict resolution method is further provided to deal with policy conflicts via balancing the need for privacy protection and the users' desire for information sharing. In addition, we provide a prototype implementation of our authorization mechanism, and our experimental results demonstrate the feasibility and usability of our approach.

The rest of the paper is organized as follows. In Section 2, we overview Google+ privacy management mechanism and analyze collaborative authorization requirements in Google+. We articulate our proposed CMAC model, including multiparty authorization specification and multiparty policy evaluation in Section 3. The details about prototype implementation and experimental results are described in Section 4. We overview the related work in Section 5. Section 6 concludes this paper and discusses our future directions.

2. PRELIMINARIES

2.1 Google+ and its Privacy Management

Google launched Google+ on June 28th, 2011, and it then attracted 25 million unique visitors after four weeks in operation [9]. This new social network platform offers many unique features different from other typical OSNs. For instance, it is able to offer a group video chat that can include up to ten people. In addition, it supports an instant messaging service for various mobile devices.

Paul Adams introduced the concept of social circles [7], which then act as the foundation of circles for user and privacy management in Google+. By default, there are four circles in Google+: "Friends", "Family", "Acquaintances" and "Following" but a user can remove/rename any of the default circles or add new circles. A user can then add any of her/his contacts to one or more circles by using a simple and intuitive drag-and-drop interface in Google+.

For privacy management, a user in Google+ can selectively share information with a *specific* set of her/his circles, *all* her/his circles, her/his *extended* circles (everyone in her/his circles plus all the people in their circles) or with the *public* (everyone). However, Google+ does not allow any *exceptions* [37]. It means if some content is shared with a larger circle, there is no way to exclude any subset of the circle.

2.2 Requirements of Collaborative Data Sharing in Google+

Users in Google+ can post notes and comments, upload photos and videos in their own spaces, tag others to their content, and share the content with their circles. On the other hand, users can also post content in others' 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 go to the bar with @Carol on Saturday night"* 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.

Google+ also enables users to *reshare* others' content. For example, when Alice views a photo in Bob's space and decides to share this photo with her circles, 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 the public, the initial privacy concerns of this photo may be violated. Thus, all access control policies defined by associated users should be enforced to regulate access of the content in disseminator's space.

In addition to *content sharing*, collaborative control is required in *circle sharing* [10] in Google+ as well, where users in a shared circle may have different privacy concerns on the circle sharing.

3. CIRCLE-BASED MULTIPARTY ACCESS CONTROL FOR GOOGLE+

To enable collaborative authorization management of data sharing in Google+, we formalize CMAC (Section 3.1), accompanying with a policy scheme (Section 3.2) and a policy evaluation mechanism (Section 3.3) for the specification and enforcement of CMAC policies in Google+.

3.1 CMAC Model

An OSN system, such as Google+, typically contains a set of users, a set of user profiles, a set of user contents, and a set of user relationships (circles in Google+). *User profile* indicates who a user is in the OSN, including identity and personal information, such as name, birthday, interests and contact information. *User content* describes what a user has in the OSN, including photos, videos, notes, events, status, and all other data objects created through various activities in the OSN. *User relationship* shows who a user knows in the OSN, representing user connections with friends, family, coworkers, colleagues, and so on.

Existing OSNs including Google+ do not provide effective mechanism to support collaborative privacy control over shared data. Several access control schemes (e.g., [19, 20, 24, 25]) have been recently introduced to support fine-grained authorization specifications for OSNs. Unfortunately, these schemes also only allow a single controller, the resource *owner*, to specify access control policies. Indeed, a flexible access control mechanism in a multi-user environment like OSNs should allow multiple controllers, who are associated with the shared data, to specify access control policies. As we discussed in Section 2.2, in addition to the *owner* (the content in the space of the user) of content, other controllers , including the *contributor* (the content published by the user in someone else's space), *stakeholder* (The tagged user associated with the content) and *disseminator* (The user shares the content from someone else's space to his/her space) of content, need to govern the access of the shared data as well due to possibly different privacy concerns. We define these controllers as follows:

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

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

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

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.*

In real life, users naturally group their connections (the people they know) into social circles, and also assign them different priorities called *trust*. Social circles and trust among connections can help a user determine how to interact with other users. The "circles" in Google+ can directly reflect the feature of social circles in real life of a user. However, the concept of "trust" cannot be explicitly represented in existing OSNs including Google+. Obviously, even users in a same circle may represent different degrees of trust, and users' trustworthiness can be also leveraged to determine who are authorized to access a resource. For example, a user may want to disclose business documents to only her/his co-workers who are with *high* trust levels. Several existing approaches [18, 20, 26, 27] have discussed how trust could be utilized in OSNs. We believe that such considerations can also apply to our collaborative privacy management scenario. Thus, in our CMAC model, we assume users can explicitly specify how much they trust others by assigning each of them a trust level when they group their connections into circles in OSNs.

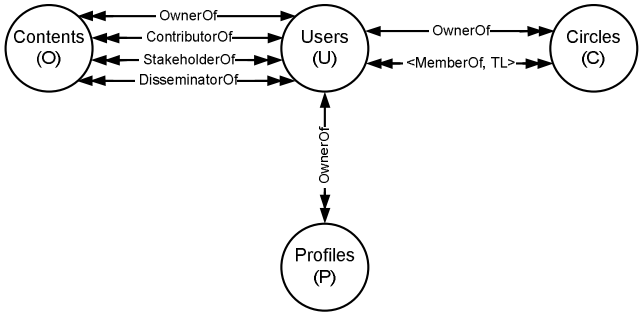

Figure 1: CMAC Model: Components and Relations.

Figure 1 represents the core components and relations of our CMAC model. Note that each content has only one *owner* and one *contributor*, and each profile item and each circle have only one *owner* as well. We now formally define this model as follows:

- $U = \{u_1, \ldots, u_n\}$ is a set of users of the OSN. Each user has a unique identifier;
- $C = \{c_1, \ldots, c_m\}$ is a set of circles created by users in the OSN. Each circle is identified by a unique identifier as well;
- $O = \{o_1, \ldots, o_p\}$ is a set of contents in the OSN. Each content also has a unique identifier;
- $P = \{p_1, \ldots, p_q\}$ is a set of user profile items in the OSN. Each profile item is a <*attribute: profile-value*> pair, $p_i =< attr_i : pvalue_i >$, where $attr_i$ is an attribute identifier and $pvalue_i$ is the attribute value;
- $UC = \{uc_1, \ldots, uc_{tr}\}$ is a collection of user circle sets, where $uc_i = \{uc_{i1}, \ldots, uc_{is}\}$ is a set of circles created by a user $i \in U$, where $uc_{ij} \in C$;
- $UP = \{up_1, \ldots, up_v\}$ is a collection of user profile sets, where $up_i = \{up_{i1}, \ldots, up_{iw}\}$ is the profile of a user $i \in U$, where $up_{ij} \in P$;
- $CT = \{OW, CB, SH, DS\}$ is a set of controller types, indicating *OwnerOf*, *ContributorOf*, *StakeholderOf*, and *DisseminatorOf*, respectively;
- $CO = \{CO_{ct_1}, \ldots, CO_{ct_x}\}$ is a collection of binary user-to-content relations, where $CO_{ct_i} \subseteq U \times O$ specifies a set of $< user, content >$ pairs with a controller type $ct_i \in CT$;
- $TL = \{tl_1, \ldots, tl_y\}$ is a set of supported trust levels, which are assumed to be in the closed interval [0,1] in our model;
- $CUT \subseteq C \times U \times TL$ is a set of 3-tuples $< circle, user, trust\_level >$ representing user-to-circle membership relations (*MemberOf*) with assigned trust levels;
- $controllers : O \xrightarrow{CT} 2^U$, a function mapping each content $o \in O$ to a set of users who are the controllers of the content with the controller type $ct \in CT$:

  $controllers(o : O, ct : CT) = \{u \in U \mid (u, o) \in CO_{ct}\}$;
- $contents : U \xrightarrow{CT} 2^O$, a function mapping each user $u \in U$ to a set of contents, where the user is a controller of the contents with the controller type $ct \in CT$:

  $contents(u : U, ct : CT) = \{o \in O \mid (u, o) \in CO_{ct}\}$;
- $user\_own\_circles : U \rightarrow 2^C$, a function mapping each user $u \in U$ to a set of circles created by this user:

  $user\_own\_circles(u : U) = \{c \in C \mid (\exists uc_u \in UC)[c \in uc_u]\}$;
- $circle\_contain\_users : C \rightarrow 2^U$, a function mapping each circle $c \in C$ to a set of users who are the members of this circle:

  $circle\_contain\_users(c : C) = \{u \in U \mid (c, u, *)$ [1] $\in CUT\}$;
- $user\_belong\_circles : U \rightarrow 2^C$, a function mapping each user $u \in U$ to a set of circles to which this user belongs:

  $user\_belong\_circles(u : U) = \{c \in C \mid (c, u, *) \in CUT\}$;
- $user\_extended\_circles : U \rightarrow 2^C$, a function mapping each user $u \in U$ to a set of circles of the user's circles:

  $user\_extended\_circles(u : U) = \{c \in C \mid (\exists u^{'} \in circle\_contain\_users(c^{'}) \wedge c^{'} \in user\_own\_circles(u))[c \in user\_own\_circles(u^{'})]\}$;
- $trust\_level : C, U \rightarrow TL$, a function returning the trust level of a user-to-circle membership relation:

  $trust\_level(c : C, u : U) = \{tl \in TL \mid (c, u, tl) \in CUT\}$;

[1] "*" is to indicate any value of the trust level within the tuple.

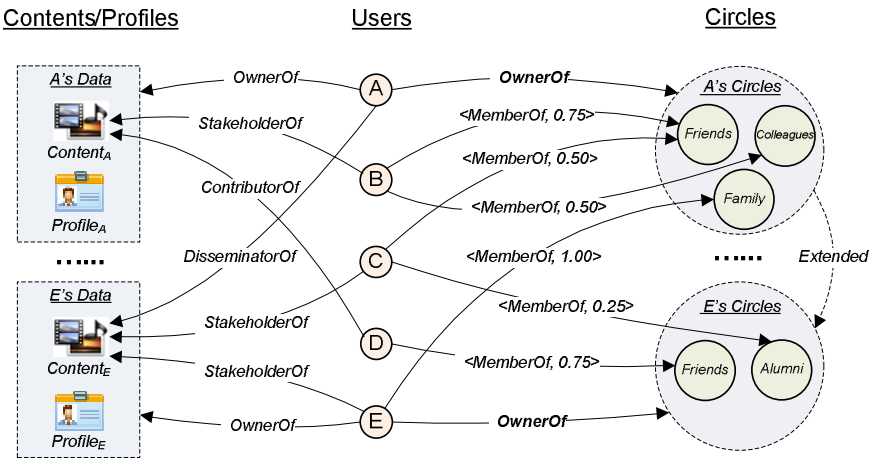

Figure 2: An Example of Circle-based Multiparty Social Network.

- $all\_circles\_users : U \rightarrow 2^U$, a function mapping a user $u \in U$ to a set of users who are the members of the user's circles:

  $all\_circles\_users(u : U) = \{u^{'} \in U \mid (\exists c \in user\_own\_circles(u))[u^{'} \in circle\_contain\_users(c)]\}$;

- $extended\_circles\_users : U \rightarrow 2^U$, a function mapping a user $u \in U$ to a set of users who are the members of the user's circles or extended circles:

  $extended\_circles\_users(u : U) = \{u^{'} \in U \mid u^{'} \in all\_circles\_users(u) \vee (\exists c \in user\_extended\_circles(u))[u^{'} \in circle\_contain\_users(c)]\}$.

Figure 2 depicts an example of circle-based multiparty social network representation. It contains five individuals, Alice (*A*), Bob (*B*), Carol (*C*), Dave (*D*) and Edward (*E*), along with their relations with user data and circles. Suppose the trust levels that a user can allocate to other users who are in her/his circles are {0.00, 0.25, 0.50, 0.75, 1.00}, indicating *none* trust, *weak* trust, *medium* trust, *strong* trust, and *strongest* trust, respectively. Note that a user may belong to more than one circle of another user and be assigned with different trust levels. For example, in Figure 2, Bob is in both "Friends" and "Colleagues" circles of Alice, and assigned with *strong* trust level and *medium* trust level, respectively. In addition, one user's circles can be the extended circles of others. For example, since Edward is in the "Family" circle of Alice, all Edward's circles are the extended circles of Alice. Moreover, this example depicts that some contents have multiple controllers. For instance, $Content_A$ has three controllers: the owner Alice, the contributor Dave and a stakeholder Bob. Also, some users may be the controllers of multiple contents. For example, Alice is the owner of $Content_A$ as well as a disseminator of $Content_E$.

3.2 CMAC Policy Specification

To achieve authorization requirements with respect to the multiparty privacy concerns, it is essential for access control policies to be in place to regulate access over shared data associated with multiple controllers. Our policy specification scheme is constructed based on the proposed CMAC model. In our model, each controller of a shared resource can specify one or more rules as her/his policy governs who can access the resource.

Accessor Specification: Accessors are a set of users who are granted to access the shared data. In Google+, accessors can be specified with a set of circles. In addition, as we discussed previously, trust levels can be used as constraints on determining authorized users in our model. We formally define the accessor specification as follows:

DEFINITION 5. *(**Accessor Specification**). Let $ac \in C \cup \{All\_Circles\} \cup \{Extended\_Circles\} \cup \{*\}$ be a specific circle $c \in C$, all circles or extended circles of the controller who defines the policy, or everyone (*) in the OSN. Let $tl_{min} \in TL$ and $tl_{max} \in TL$ be, respectively, the minimum trust level and the maximum trust level that the users in ac must have. The accessor specification is defined as a set, $\{a_1, \ldots, a_n\}$, where each element is a tuple $< ac, tl_{min} >$ for* positive *rule (with "permit" effect) or $< ac, tl_{max} >$ for* negative *rule (with "deny" effect).*

Data Specification: In Google+, users can share their contents, profiles, even circles with others. To facilitate effective policy conflict resolution for multiparty access control (Section 3.3.1), we introduce *sensitivity levels* for data specification, which are assigned by the controllers to the shared data. A user's judgment of the sensitivity level 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 6. *(**Data Specification**). Let $dt \in O \cup C \cup P$ be a data item. Let sl be a sensitivity level, which is a rational number in the range [0,1], assigned to dt. The data specification is defined as a tuple $< dt, sl >$.*

Access Control Policy: To summarize the above-mentioned policy elements, we give the definition of CMAC access control rule as follows:

DEFINITION 7. *(**CMAC Rule**). A CMAC rule is a 5-tuple $R =< controller, ctype, accessor, data, effect >$, where*

- *$controller \in U$ is a user who can regulate the access of $data$;*
- *$ctype \in CT$ is the type of the controller;*
- *$accessor$ is a set of users to whom the authorization is granted, representing with an access specification defined in Definition 5.*

- *data is represented with a data specification defined in Definition 6; and*

- *effect* $\in \{permit, deny\}$ *is the authorization effect of the rule.*

Note that the semantics of accessor specification, $\{a_1, \ldots, a_n\}$, in a rule can be explained as the *conjunction* of elements in accessor specification, $a_1 \wedge \ldots \wedge a_n$, which means that only *common* users in the accessor sets defined by the elements in accessor specification are treated as authorized users. Suppose 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 rule:

EXAMPLE 1. *Alice authorizes users who are in both her "Friends" circle and her "Colleagues" circle with at least a medium trust level to access a photo named funny.jpg she is tagged in, where Alice considers the photo with a high sensitivity level and she is a stakeholder of the photo:*
$r_1 = (Alice, SH, \{< Friends, 0.50 >, < Colleagues, 0.50 >\}, < funny.jpg, 0.75 >, permit)$.

Applying this rule to the example social network shown in Figure 2, Bob satisfies this rule, since he is in both "Friends" and "Colleagues" circles of Alice with the trust levels greater than and equal to the trust threshold defined in the rule.

Furthermore, one controller may define more than one rule in her/his policy for a shared resource. In this case, users who satisfy *any* rule in the policy are considered as authorized users for the resource. The following is another example rule:

EXAMPLE 2. *In addition to the rule defined in Example 1, let's consider another authorization requirement from Alice, where she also wants to disclose the same photo to users in her "Family" circle with any trust level:*
$r_2 = (Alice, SH, \{< Family, 0.00 >\}, < funny.jpg, 0.75 >, permit)$.

Finally, this is a common requirement that controllers can exclude specific groups from the authorized users when they define the policies for shared data. Our policy scheme supports such an *exception* feature through the definition of positive rules in access control policies [2].

EXAMPLE 3. *Suppose Alice wants to share one of her videos, party.avi, with users who are in her "Friends" circle but disallows users in her "Colleagues" circle to watch this video, where Alice considers the video with a medium sensitivity level and she is the owner of the video. Then, Alice can define her policy with following two rules for the video:*
$r'_1 = (Alice, OW, \{< Friends, 0.00 >\}, < party.avi, 0.50 >, permit)$*; and*
$r'_2 = (Alice, OW, \{< Colleagues, 1.00 >\}, < party.avi, 0.50 >, deny)$.

When we apply this policy to the example social network (Figure 2), although both Bob and Carol are in Alice's "Friends" circle, only Carol can see the video since Bob is also in Alice's "Colleagues" circle that is excluded by the r'_2.

[2]Note that both *conjunction* and *exception* features in our policy specification scheme cannot be supported by the current privacy setting mechanism in Google+.

3.3 CMAC Policy Evaluation

In our CMAC model, we adopt three steps to evaluate an access request over multiparty access control policies as shown in Figure 3. The first step checks the access request against the policy specified by each controller and yields a decision for the controller. In our CMAC model, controllers can leverage a positive rule to define a set of circles to whom the shared resource is visible, and a negative policy to exclude some specific circles from whom the shared resource should be hidden. A strategy called `deny-overrides`, which indicates that "deny" rule take precedence over "permit" rule, is adopted to achieve such an exceptional feature in our policy evaluation mechanism. In the second step, decisions from all controllers responding to the access request are aggregated to make a *collaborative* decision for the access request. Since those controllers may generate different decisions (`permit` and `deny`) for the access request, conflicts may occur again. We will address our approach for resolving such conflicts in details subsequently. In addition, if the target of the access request is a resource disseminated by a disseminator, the third step is needed for policy evaluation. In this case, the disseminator may specify a conflicting privacy control over the disseminated content with respect to the original controllers of the content. In order to eliminate the potential disclosure risk of sensitive information from the procedure of data dissemination, we again leverage the restrictive conflict resolution strategy, `Deny-overrides`, to resolve conflicts between original controllers' decision and the disseminator's decision.

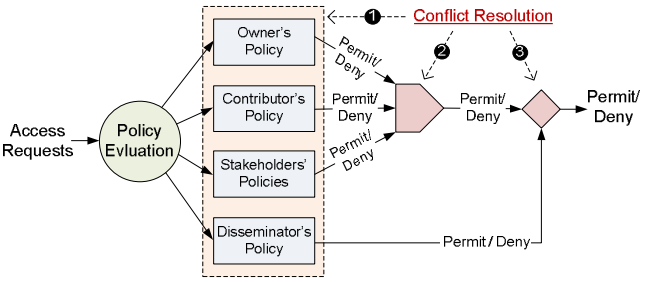

Figure 3: CMAC Policy Evaluation Process.

3.3.1 Conflict Resolution in Multiparty Control

The essential reason caused the conflicts among multiple controllers is that each controller of the shared content often has a different privacy concern over the content. For example, assume that Alice and Bob are two controllers of a photo. Both of them define their own access control policy stating only her/his friends can view this photo. Since it is almost impossible that Alice and Bob have the same set of friends, privacy conflicts may always exist when considering multiparty control over the shared data item. The process of privacy conflict resolution makes a decision to allow or deny the requestor to access the shared content. In general, allowing a requestor to access the content may cause *privacy risk*, but denying a requestor to access the content may result in *sharing loss*. Our privay conflict resolution approach attempts to find a tradeoff between privacy protection and data sharing.

Measuring Privacy Risk: The privacy risk of an access request is an indicator of potential threat to the privacy of controllers in terms of the shared content: the higher the privacy risk of an access request, the higher the threat to controllers' privacy. Our basic premises for the measurement of privacy risk for an access request are the following: (a) the lower the trust levels of the requestor

who requires the access request, the higher the privacy risk; (b) the lower the number of controllers who allow the requestor to access the content, the higher the privacy risk; (c) the stronger the general privacy concerns of controllers, the higher the privacy risk; and (d) the more sensitive the shared data item, the higher the privacy risk. Therefore, the privacy risk of a conflicting segment is calculated by a monotonic function with the following parameters:

- **Trust of the requestor**: The trust level of the accessor i is denoted as tl_i, which is the *average* of the trust levels of the accessor with respect to all controllers of the shared content. The trust level of an accessor who belongs to a circle of a controller is defined by the controller directly. The trust level of an indirect relationship, which is represented by a relationship chain linking an accessor and a controller, needs to be calculated based on the trust levels associated with the relationships composing the chain. Several algorithms for computing the trust of indirect relationships in OSNs have been proposed in [13, 20, 26, 30]. The discussion of selecting effective algorithm for calculating the trust of indirect relationships is beyond the scope of this paper; [3]
- **Number of denied controllers** : The denied controllers are controllers who make the "deny" decisions for the access request. All denied controllers of an access request i are returned by a function $controllers_d(i)$;
- **General privacy concern of a denied controller**: The general privacy concern of a denied 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 group of data. For example, public figures may have higher privacy concern on their shared photos than ordinary people; and
- **Sensitivity of the content**: Content sensitivity defines controllers' perceptions of the confidentiality of the content being transmitted. The sensitivity level of the shared content explicitly chosen by a denied controller j is denoted as sl_j. The factor depends on the denied controllers themselves. Some denied controllers may consider the shared content with a higher sensitivity.

In order to measure the privacy risk of an accessor i, denoted as PR(i), we can use following equation to aggregate the privacy risks of i due to different denied controllers.

$$PR(i) = (1 - tl_i) \times \sum_{j \in controllers_d(i)} pc_j \times sl_j \qquad (1)$$

Measuring Sharing Loss: When the decision of privacy conflict resolution for an access request is "deny", it may cause losses in potential content sharing, since there are controllers expecting to allow the requestor to access the data item. Similar to the measurement of the privacy risk, four factors are adopted to measure the sharing loss for a requestor. Compared with the factors used for quantifying the privacy risk, the difference is that we only consider *allowed controllers* for evaluating the sharing loss of an accessor. The sharing loss $SL(i)$ of an accessor i is the aggregation of sharing loss with respect to all allowed controllers as follows:

$$SL(i) = tl_i \times \sum_{k \in controllers_a(i)} (1 - pc_k) \times (1 - sl_k) \qquad (2)$$

[3]We adopt a trust computation algorithm discussed in [20] to compute the trust of indirect relationships in the current version of our system.

where, function $controllers_a(i)$ returns all allowed controllers of a requestor i.

Conflict Resolution: The tradeoff between privacy and utility in data publishing has been recently studied [17, 32]. Inspired by those work, we introduce a mechanism to balance privacy protection and data sharing for an effective privacy conflict resolution in OSNs.

We first calculate the privacy risk ($PR(i)$) and the sharing loss ($SL(i)$) for each requestor (i), individually. Then, following equation can be utilized to make the decisions (permitting or denying an access request) for privacy conflict resolution.

$$Decision = \begin{cases} \text{Permit} & \text{if } \alpha SL(i) \geq \beta PR(i) \\ \text{Deny} & \text{if } \alpha SL(i) < \beta PR(i) \end{cases} \qquad (3)$$

where, α and β are preference weights for the privacy risk and the sharing loss, $0 \leq \alpha, \beta \leq 1$ and $\alpha + \beta = 1$.

4. IMPLEMENTATION AND EVALUATION

4.1 Prototype System Implementation

We implemented a proof-of-concept social network application to demonstrate collaborative management of photos, called *Sigma* after the symbol used for multiple addition (or plus) operations. The intent of the application is to allow users to collaboratively share photos in Google+ based on our approach. However, constrained by current lack of development API for Google+, our implementation is a Facebook application [4] using Facebook users' data to simulate an environment like Google+.

Sigma is developed as a third-party Facebook application which is hosted on an Apache Tomcat server supporting PHP and MySQL, with a user interface built using HTML/CSS/JavaScript, jQuery, and jQuery UI. *Sigma* is built using Facebook API calls to integrate users' photos and friend lists on a stand-alone application ultimately intended to mimic Google+ functionality.

Figure 4 shows the architecture of *Sigma*. The application is hosted on an external web server, but uses Facebook's graph API and Facebook Query Language to retrieve user data. A minimal amount of data is kept on the server itself, but our application allows users to save their settings and check access to their photos based on the result of the multiparty policy evaluation.

Sigma consists of two major parts, a circle management module and a photo management module. The circle management module, shown in Figure 5 (d), allows users to sort their friends into circles based on their existing Facebook friend lists. It also allows them to set trust levels by friend or by circle. For the performance purpose in using the application, setting the trust level for a circle applies it to all individual users in that circle in our current implementation. In a real-life implementation, the function of circle trust level would depend on the type of circle. If it is a trust-based circle, trust level may be used as an indication of which users to place in that circle. If it is a group-based circle, it might display an average trust level of all the users.

In the photo management module of *Sigma*, the privacy setting depicted in Figure 5 (b) accounts for the three features: *union*, *conjunction*, and *exception*, discussed in our policy specification scheme. To achieve this, three options are presented and then joined by union for the ultimate policy. The controller indicates a set of circles and/or users who may access the photo, a set of circles of which the intersection of users may access the photo, and a set of circles and/or users who may not access the photo. The controller

[4]The link of our application is removed for anonymous review.

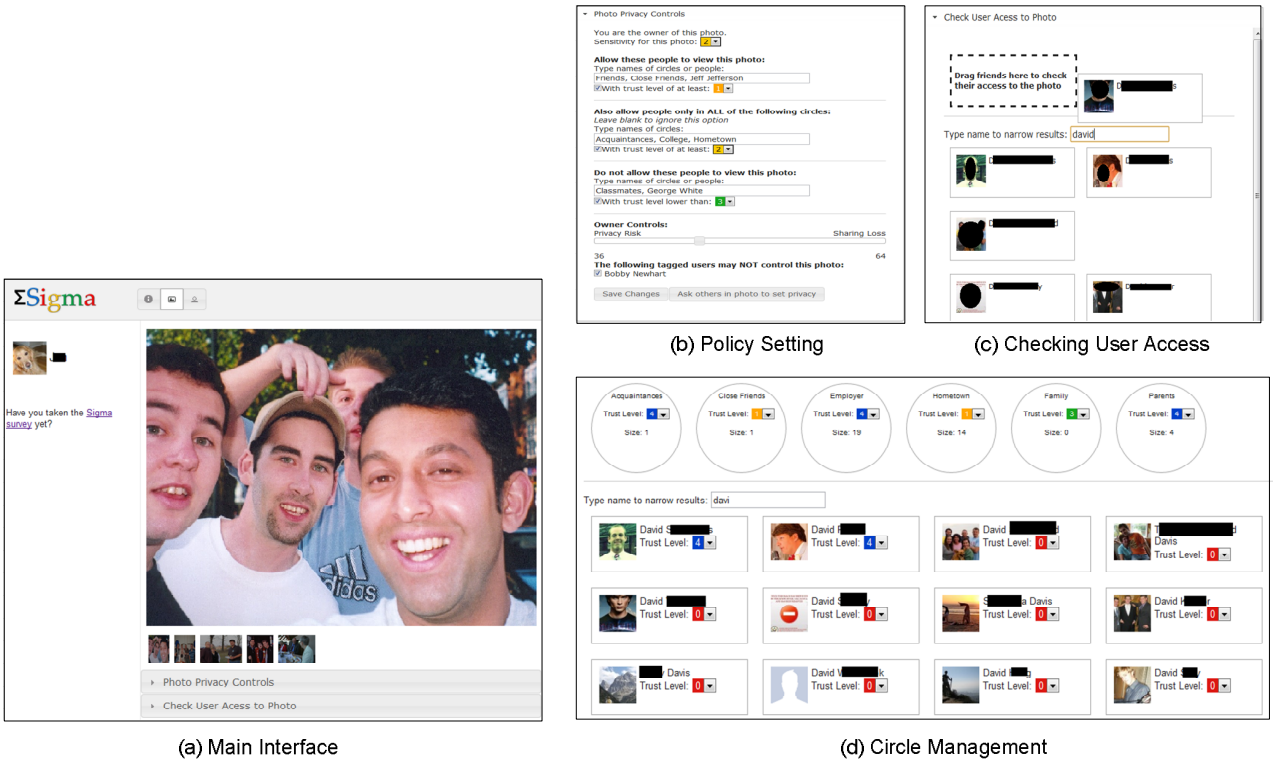

Figure 5: *Sigma* Interfaces.

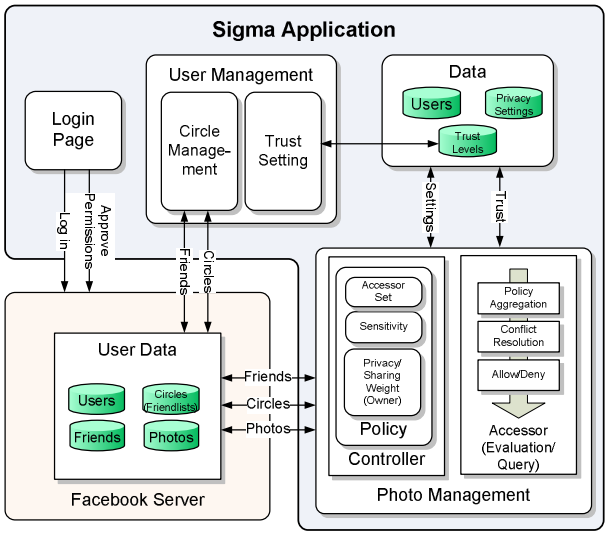

Figure 4: System Architecture of *Sigma*.

may also optionally indicate a minimum trust level for a "*permit*" policy or a maximum trust level for a "*deny*" policy to additionally restrict photo sharing. If the controller is the owner of selected photo, s/he can adjust the weights to balance privacy protection and data sharing of the photo. In addition, since *malicious* users may tag themselves to a photo and specify privacy policies to influence the sharing of the photo, the photo owner can verify the tagged users and has the ability to disable fake stakeholders to control the photo in the privacy setting. To allow the users of the prototype application to check the impacts of collaborative control against their privacy settings, users are able to check friends' access to the photo in *Sigma* as shown in Figure 5 (c).

We chose to supply the user with five photos for the purposes of evaluation, so that we could ask reasonably users to set privacy settings for all of the photos displayed. The photo management module in *Sigma* automatically selects five photos the user is tagged in, giving preference to photos with more people tagged (to promote collaborative use of the application), shown in Figure 5 (a). The user can then use the photo privacy controls or check user access to the photo.

4.2 Prototype System Evaluation

4.2.1 User Study

We conducted a usability study to test the feasibility of *Sigma*. We had 42 users use the application and answer a survey to indicate their preferences in social networks. We recruited through University mailing lists, Google+ and Facebook. Of our respondents, 71.4% were 18-24, 21.4% were 25-34, and 7.1% were 35-54 years old. Some questions were "ranking" questions, where users were asked to rank certain things by preference. Responses were then assigned a weight of (`n-r`) where `n` is the total number of data items to rank and `r` is the rank assigned. Therefore, rating something 3 out of 5 gives a score of 2. Responses from all users are then totaled for comparison. The details of responses are compiled in the Appendix.

Prior to using Sigma: Part of the purpose of the survey was to understand the demand for a collaborative data management system that balances privacy protection with data sharing. When asked whether privacy or sharing was more important, half of respondents rated them as equally important (with 32.1% finding choosing pri-

vacy and 17.9% choosing sharing), so we know both are necessary when determining an approach to data management in OSNs.

When asked to rank preferences when tagged in a photo, users indicated that protecting their privacy was the most important to them (a score of 79), with sharing with friends and protecting other users' privacy were ranked closer to each other (53 and 41, respectively). Asked to rank preferences when a user owns a photo, they indicated protecting their own privacy and sharing closely (92 and 81), with protecting tagged users' privacy (65) still somewhat important and allowing tagged users to share with their friends (42) last.

When a user is tagged, we can see that protecting their own privacy is important. Since in a normal social network a tagged user has little protection compared to the owner, we can interpret this as a desire for more control over tagged photos, since the current approach allows the owner to override control. When a user owns a photo, they consider privacy protection and sharing loss about equal, but they consider protecting tagged users' privacy important as well (a score of 65 indicates that some users ranked it as at least the 2nd most important).

We also asked the users who use Google+ (64%) about how they sort their circles to analyze the value of group-based and trust-based circles (Table 1). Most of users said they organize by friendship (how well they know someone), and users organize friends by interest (group) and by trust about equally. Since a large portion of users already sort users by group or by trust, it would be beneficial to incorporate this in collaborative privacy control.

Table 1: Factors Used When Sorting Users into Circles.

What factors do you use when sorting other users into circles?	% of respondents
Oragnize friends by interest	44.4%
Organize friends by friendship	61.1%
Oragnize friends by trust	44.4%
Organize friends for sharing control not related to trust or interests	16.7%

After Using Sigma: We collected some Facebook usage statistics to determine the need for collaborative photo management. We define need as the presence of more than one party interested in a photo (the number of controllers is greater than one). We can estimate from the data that, in owned photos, there is on average at least two tagged users for every five photos. More importantly, about 15% of owned photos have at least two tagged users, and about 5% have three or more. This means in an *only-owner-control* approach for privacy management, a sizable number of users is being ignored in determining privacy settings for those photos.

In photos in which our test subjects are tagged, there are an average of two users in every photo. This means in the *only-owner-control* approach, we can assume that there is at least one user for every tagged photo whose privacy preferences are not being taken into account. In addition, 25% of these photos have more than two tagged users. This high amount of stakeholders in various photos supports the need for a collaborative privacy approach.

We also asked users to rank their preferences for various parts of our system as they tried it out. For a user management system, users ranked their preferences as shown in Table 2. Users ranked the ability to indicate trust almost as important as simplicity, meaning they reacted very positively to this feature of our system. The other scores indicate the desire for intelligent sorting of friends, as well as the importance of having a visual interface (like circles).

We then again asked users to rank preferences in sharing, but for three scenarios: when the user is a stakeholder, when the user is an owner, and in general when collaboratively controlling a photo (Table 3). In all three situations, the user ranked protecting one's own privacy as the most important. This may seem obvious, but it is important to note that this suggests they find protecting one's privacy as a stakeholder equally important to protecting one's privacy as an owner (supporting the need for collaborative control). Users indicated that when they were tagged, having an equal say to the owner was least important, so if the owner has more control in the system (such as setting weights in our system) it is permissible as long as the stakeholders have a say. In general and as an owner, users indicated that owner control was second-most important, which further supports the need for some additional owner controls like ours in a collaborative approach. When a user owns a photo, the third highest preference, above sharing and giving tagged users control, was preventing fake tagged users from controlling a photo. Besides the design concern of preventing collusion attacks as well as other threats from false tagging, users indicated that indication of *valid* controllers was an important feature to them.

Table 2: Importance of Features in User Management.

Rate the features of this or a similar user management system in order of importance	Weighted Score
Simplicity	146
Ability to indicate trust	115
Automatically sorting friends	93
Visual interface	90
Recommending trust levels for friends	76
Recommending circle placement	68

Table 3: Importance of Features in Collaborative Sharing.

Rate the following in order of importance when collaboratively sharing a photo	Weighted Score
Tagged	
Protecting my privacy	99
Ability to prevent users from viewing photo	83
Ability to allow users to view photo	76
Sharing	59
Having an equal say to the owner	58
Owned	
Protecting my privacy	95
Having complete control	89
Preventing fake tagged users from controlling	72
Sharing	61
In General	
Protecting privacy	80
Giving the owner control	72
Giving tagged users control	52
Allowing users to share	46

We asked users to rank the reasons they would use the "Check Access" function to determine the need for the function. Users ranked as follows: making sure certain people cannot access a photo (121), to double-check privacy settings on a photo (101), to make sure certain people can access a photo (88), to see how other controllers' choices affect my privacy (70), and because they do not understand how the settings work (32). It makes sense that users would mostly want to make sure a specific person should not access a photo due to their concern about privacy, but this also indicates the usefulness of such a function. What is important to note is that a score of 32 is low enough that it indicates most users rated this as the least likely reason. Users are not worried about misunderstanding how their settings work and are simply concerned with protecting themselves, so our system is not prohibitively complicated.

Obviously a privacy system is only useful if it is, in fact, used. Shown in Table 4, we asked users how often they would use collaborative control for photos. 80% of users would collaboratively control photos they are tagged in, which is sizable enough to jus-

tify collaborative control from a user-centric standpoint. However, 44% of users indicate they would collaboratively control most or all of their individual photos. This means that even with a different system than they are used to, over 2 out of every 5 users would use a system like this for a majority of their photos.

Table 4: Frequency and Usage of Collaborative Photo Privacy.

Which of the following would you use collaborative privacy for?	% of respondents
Sensitive Photos	64%
Specific Albums	68%
Most or all individual photos	44%
Photos I own	60%
Photos I am Tagged in	80%

4.2.2 *Effectiveness Evaluation*

To evaluate the effectiveness of our approach, we compare the outcome, on a single-accessor basis, of a policy set in Google+ to a policy set in *Sigma*. The metric we use for evaluation is the total Privacy Risk (`PR`) plus the total Sharing Loss (`SL`) from all controllers based on the outcome of the access attempt.

We evaluate the outcome in a few cases. The outcome is a measurement of average expected privacy risk and sharing loss (which uses average trust levels and average sensitivity levels). It should be noted, however, that higher trust or lower sensitivity would simply lower the magnitude of the final measurements and lower trust or higher sensitivity would simply increase the magnitude of the final measurements, but the comparison still holds. Additionally, since we are evaluating on a single-accessor basis, number of friends or circles allowed or denied do not affect the results.

One case is trivial: in both Google+ and *Sigma*, if all users agree on the same privacy setting, there are no conflicts to resolve. The result is 0 `PR` and 0 `SL` in either Google+ or *Sigma*. This is considered the best case. The rest of the cases and evaluation results are shown in Figure 6.

Case 1 is in Google+ (or any owner-override situation) where all of the stakeholders in a photo disagree with the owner. This is a worst-case for Google+. This can be compared with Case 6, which is the same access decision in *Sigma*. In Google+ the privacy risk or sharing loss grows with each non-owner controller, as his or her decision is being violated. In *Sigma*, this is only slightly different from the best-case scenario. In Cases 2-5, half of the stakeholders agree with the owner. In Case 2, the owner allows in Google+ and in Case 3 the owner denies in *Sigma*. In Case 4 the owner denies in Google+ and in Case 5 the owner allows in *Sigma*. This can be considered an "average case". In these cases, *Sigma*'s scores increase at the same rate as Google+. This shows that *Sigma* is at least as good as Google+, until one considers the fact that this "average case" for Google+ is actually the worst case for *Sigma*.

It is important to note that the rate of `PR` or `SL` as number of controllers increases is at most 1/2 in *Sigma*. This is due to the fact that the maximum proportion of controllers whose preferences are being violated is 1/2, since (given the same sensitivity and trust settings) more than 50% controllers in agreement determine the decision. In Google+, this is not the case. In fact, `PR` or `SL` will increase for every new controller who disagrees with the owner since the decision is never changed. This is why Cases 2 and 4 increase at the same rate as *Sigma*'s maximum rate in Cases 3 and 5 – every second controller disagrees with the owner. Thus, *Sigma*'s worst case is at least as effective at giving user preference as Google+ and can only be better in other cases.

4.2.3 *Performance Discussion*

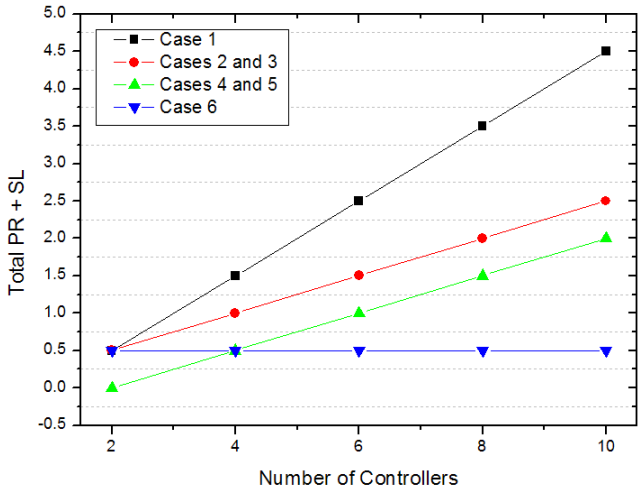

Figure 6: Privacy Risk and Sharing Loss in Google+ and *Sigma* in Six Cases – Case 1: all stakeholders in Google+ disagree with owner; Case 2: half of stakeholders in Google+ disagree with owner, who allows access; Case 3: half of stakeholders in *Sigma* disagree with owner, who denies access; Case 4: half of stakeholders in Google+ disagree with owner, who denies access; Case 5: half of stakeholders in *Sigma* disagree with owner, who denies access; and Case 6: all stakeholders in *Sigma* disagree with owner.

Though we measured the amount of time of certain operations within our system, there was a high variability based on the status of the shared web server and the speed of responses from Facebook API calls, so our discussion of performance would revolve around the scalability of our system.

Our implementation retains user privacy policies, then evaluates these policies when each time an access attempt is made. This has the advantage of only computation-centric access as needed, versus building an accessor list for shared data every time when a privacy policy is created [5]. This means collaborative changes can be made at a very low performance cost to the system. There are some drawbacks to the compute-on-access approach, though. A highly-accessed data item may decrease system performance. Additionally, this implementation makes checking the access for a group of users decidedly inefficient, considering policies from multiple controllers. For instance, *Sigma*'s "Check Access" function applied to an entire circle depends on evaluating policies once per user. In the approach that builds an accessor list every time when a privacy policy is changed, checking access for a group of individuals depends only on the speed of a single database look-up. The access performance could be improved with a cache, providing the result of an accessor's last attempt to view a data item and a "dirty bit" to check if the policy for the photo has changed.

The key advantage of this implementation is that it provides for scalability in almost every area other than frequency of access attempts. The computation of a policy does not strongly depend on the size of the set of allowed circles and accessors a user indicates. Instead, evaluation of a policy simply checks if the current accessor is in that set, which is significantly more efficient than building an accessor list. Thus, as number of users, number of circles, and number of friends in those circles increase, policy evaluation effi-

[5]In this case, all policy conflicts need to be resolved first for building the accessor list.

ciency depends only on the number of accesses to a photo. Another performance advantage is the space complexity compared to building an accessor list. As the number of users, circles, and friends increases, the space complexity of an accessor list does as well.

5. RELATED WORK

Several access control models for OSNs have been proposed recently [19, 20, 24, 25, 30]. The D-FOAF system [30] is primarily a Friend of a Friend (FOAF) ontology-based distributed identity management system for OSNs, where relationships are associated with a trust level, which indicates the level of friendship between the users participating in a given relationship. Then, Carminati et al. [19] introduced a conceptually-similar but more comprehensive trust-based access control model, 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 also introduced a semi-decentralized discretionary access control model and a related enforcement mechanism for controlled sharing of information in OSNs [20], and proposed a semantic web based access control framework for social networks. Fong et al. [25] presented an access control model that formalizes and generalizes the access control mechanism implemented in Facebook, admitting arbitrary policy vocabularies that are based on theoretical graph properties. Gates [21] claimed relationship-based access control as one of new security paradigms that addresses unique requirements of Web 2.0. Then, Fong [24] formulated this paradigm called a Relationship-Based Access Control (ReBAC) model that bases authorization decisions on the relationships between the resource owner and the resource accessor in an OSN. However, most of these existing work could not model and analyze access control requirements with respect to collaborative authorization management of shared data in OSNs.

Recently, semantic web technologies have been used to model and express fine-grained access control policies for OSNs (e.g., [16, 22, 38]). Especially, Carminati et al. [16] proposed a semantic web based access control framework for social networks. Three types of policies are defined in their framework, including authorization policy, filtering policy and admin policy, which are modeled with the Web Ontology Language (OWL) and the Semantic Web Rule Language (SWRL). Access control policies regulate how resources can be accessed by the participants; filtering policies specify how resources have to be filtered out when a user fetches an OSN page; and admin policies can determine who is authorized to specify policies. Although they claimed that flexible admin policies are needed to bring a system to a scenario where several access control policies specified by distinct users can be applied to the same resource, the lack of formal descriptions and concrete implementation of the proposed approach leaves behind the ambiguities of their solution.

The need of collaborative management for data sharing, especially photo sharing, in OSNs has been addressed by some recent research [15, 28, 31, 40, 44]. Particularly, Squicciarini et al. [40] 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, 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 this approach indicates only the winning bids could determine who was able to access the data, instead of accommodating all stakeholders' privacy preferences. In contrast, we propose a formal model to support the multiparty access control for OSNs, along with a policy specification scheme and a simple but flexible conflict resolution mechanism to particularly enable collaborative data sharing in Google+.

Several recent work also explored how to measure privacy risk in OSNs [14, 34, 43]. Becker et al. [14] presented *PrivAware*, a tool to detect and report unintended information loss through quantifying privacy risk associated with friend relationship in OSNs. Liu et al. [34] introduced a framework to compute the privacy score of a user, indicating the user's potential risk caused by her/his participation in OSNs. Those prior solutions only focused on the privacy settings of users with respect to their profile items. However, our approach measures the privacy risk caused by different privacy concerns from multiple users, focusing on content sharing in Google+.

A very preliminary analysis of Google+ privacy has been discussed in [37]. They only addressed concern that Google+ shares the metadata of the photos uploaded by its users, which could lead to privacy violations, and also showed that Google+ encourages their users to provide other names, such as maiden names, which may help in identity theft.

6. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel mechanism for collaboratively controlling the shared data in Google+. A circle-based multiparty access control model has been formulated, along with a policy specification scheme and corresponding policy evaluation mechanism. A proof-of-concept implementation of our solution called *Sigma* and the usability evaluation of our approach have been discussed as well.

As part of our future work, we will implement and evaluate our approach in Google+ platform once Google releases the Google+ application development API. In addition, since a Google+ user may be involved in the control of hundreds of circles, the management of circles and the configuration of the privacy preferences based on circles are time-consuming and tedious tasks. Thus, we would study inference-based techniques [23, 29, 41] for both smarter circle management and automatic configuration of privacy preferences in Google+. Besides, we will extend our work to address security and privacy challenges for emerging information sharing services such as location sharing [3].

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 Privacy Policy. http://www.facebook.com/policy.php/.

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

[3] Google Latitude. www.google.com/mobile/latitude/.

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

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

[6] Facebook Friend Lists, 2007. http://www.facebook.com/blog.php?post=7831767130.

[7] P.adams. The real life social network v2, 2010. http://www.slideshare.net/padday/the-real-life-social-network-v2.

[8] The first google+ privacy flaw, 2011. http://blogs.ft.com/fttechhub/2011/06/google-plus-privacy-flaw/#axzz1cxeoa9LS.

[9] Google+ hits 25 million visitors, gets more sticky, 2011. http://mashable.com/2011/08/02/google-plus-25-million-visitors/.

[10] Google+ shared circles are here, 2011. http://www.zdnet.com/blog/google/google-shared-circles-are-here/3355.

[11] The State of Social Media 2011: Social is the new normal, 2011. http://www.briansolis.com/2011/10/state-of-social-media-2011/.

[12] F. Adu-Oppong, C. Gardiner, A. Kapadia, and P. Tsang. Social circles: Tackling privacy in social networks. In *Symposium on Usable Privacy and Security (SOUPS)*. Citeseer, 2008.

[13] P. Avesani, P. Massa, and R. Tiella. A trust-enhanced recommender system application: Moleskiing. In *SAC*, volume 5, pages 1589–1593. Citeseer, 2005.

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

[15] 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.

[16] S. Brands. *Rethinking public key infrastructures and digital certificates: building in privacy*. The MIT Press, 2000.

[17] 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.

[18] B. Carminati, E. Ferrari, S. Morasca, and D. Taibi. A probability-based approach to modeling the risk of unauthorized propagation of information in on-line social networks. In *Proceedings of the first ACM conference on Data and application security and privacy*, pages 51–62. ACM, 2011.

[19] 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.

[20] 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.

[21] 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.

[22] N. Elahi, M. Chowdhury, and J. Noll. Semantic access control in web based communities. In *Computing in the Global Information Technology, 2008. ICCGI'08. The Third International Multi-Conference on*, pages 131–136. IEEE, 2008.

[23] L. Fang and K. LeFevre. Privacy wizards for social networking sites. In *Proceedings of the 19th international conference on World wide web*, pages 351–360. ACM, 2010.

[24] 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.

[25] 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.

[26] J. Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, University of Maryland, Collega Park, 2005.

[27] J. Golbeck. Trust and nuanced profile similarity in online social networks. *ACM Transactions on the Web (TWEB)*, 3(4):1–33, 2009.

[28] H. Hu, G.-J. Ahn, and J. Jorgensen. Detecting and resolving privacy conflicts for collaborative data sharing in online social networks. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC'11. ACM, 2011.

[29] S. Jones and E. O'Neill. Contextual dynamics of group-based sharing decisions. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 1777–1786. ACM, 2011.

[30] 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.

[31] 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.

[32] 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.

[33] H. Lipford, A. Besmer, and J. Watson. Understanding privacy settings in facebook with an audience view. In *Proceedings of the 1st Conference on Usability, Psychology, and Security*, pages 1–8. USENIX Association Berkeley, CA, USA, 2008.

[34] 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.

[35] Y. Liu, K. Gummadi, B. Krishnamurthy, and A. Mislove. Analyzing Facebook Privacy Settings: User Expectations vs. Reality. In *Proceedings of the 2011 annual conference on Internet measurement (IMC'11)*. ACM, 2011.

[36] 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.

[37] S. Mahmood and Y. Desmedt. Poster: preliminary analysis of google+'s privacy. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 809–812. ACM, 2011.

[38] A. Masoumzadeh and J. Joshi. Osnac: An ontology-based access control model for social networking systems. *IEEE International Conference on Privacy, Security, Risk and Trust*, 0:751–759, 2010.

[39] F. Ozenc and S. Farnham. Life "Modes" in Social Media. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 561–570. ACM, 2011.

[40] 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.

[41] A. Squicciarini, S. Sundareswaran, D. Lin, and J. Wede.

A3p: adaptive policy prediction for shared images over popular content sharing sites. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 261–270. ACM, 2011.

[42] K. Strater and H. Lipford. Strategies and struggles with privacy in an online social networking community. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1*, pages 111–119. British Computer Society, 2008.

[43] 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.

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

[45] Y. Wang, S. Komanduri, P. Leon, G. Norcie, A. Acquisti, and L. Cranor. I regretted the minute I pressed share": A qualitative study of regrets on Facebook. In *Symposium on Usable Privacy and Security*, 2011.

APPENDIX

A. USER STUDY TABLES

Table 5: Age of Respondents.

Age	% of Respondents
18-24	21.4%
25-34	61.1%
35-54	7.1%

Table 6: Education Completed by Respondents.

Highest Level of Education Completed	% of Respondents
Graduated HS or Equivalent	3.6%
Some college, no degree	25%
Bachelor's Degree	3.6%
Bachelor's Degree (in progress)	39.3%
Post-Graduate degree	10.7%
Post-Graduate degree (in progress)	17.9%

Table 7: Social Networks Used by Respondents.

Social Network Used	% of Respondents
Facebook	100%
Google+	64.3%
Twitter	32.1%

Table 8: Time Respondents Spend on Social Networks.

Hours per week spent on Social Networking Sites	% of Respondents
Less than 1 Hour	3.6%
1-5 hours	42.9%
5-10 Hours	32.1%
10+ Hours	21.4%

Table 9: Privacy and Sharing Preferences of Respondents.

Which is more important to you when using a social network?	% of Respondents
Protecting my privacy	32.1%
Sharing	17.9%
About the same	50%

Table 10: Preferences when User Is Tagged.

Rank the following by importance to you when you are tagged in a photo	Weighted Score
Protecting my privacy	74
Sharing with friends	53
Protecting other tagged users' privacy	41

Table 11: Preferences when User Owns a Photo.

Rank the following by importance when you own a photo others are tagged in	Weighted Total
Protecting my own privacy	92
Sharing with my friends	81
Protecting tagged users' privacy	65
Allowing Tagged users to share photo with their friends	42

Table 12: Usage of Access Checking Function.

Rank the following reasons by likelihood of reason for using this function	Weighted Score
Make sure certain people can't access a photo	121
Double-check privacy settings on a photo	101
Make sure certain people can access a photo	72
To see how other controllers' choices affect my privacy	58
Because I don't understand how the settings work	32

Table 13: Photos Respondent would Collaboratively Control on Social Networks.

Which photos would you collaboratively control on a social network?	% of respondents
Sensitive photos	64%
Specific albums	68%
Most or all individual photos	44%
Photos I own	60%
Photos I am tagged in	80%