Enabling Collaborative Data Sharing in Google+

Hongxin Hu
Delaware State University,
Dover, Delaware, 19901
hxhu@asu.edu

Gail-Joon Ahn and Jan Jorgensen
Arizona State University,
Tempe, Arizona, 85287
{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 extend and formulate a multiparty access control model, named MPAC+, 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.**

I. Introduction

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. However, studies have consistently shown that users struggle to adopt this feature for managing their friends and customizing their privacy settings [10]. To address such an issue, Google recently launched a new social network service, namely Google+, 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 [9].

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. In another example, the first privacy flaw in Google+ was identified in [1] 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 multiparty access control 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., [2], [3]). In particular, we introduce the notions of *circle* and *trust* into our model, which significantly extends our multiparty authorization framework for Facebook-style social networks [5], [7]. In addition, our model contains a multiparty policy specification scheme, as well as a policy evaluation mechanism, which deals with policy conflicts by keeping the balance between the need for privacy protection and the users' desire for information sharing. Moreover, 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 II, we articulate our proposed MPAC+ model, including MPAC+ policy specification and MPAC+ policy evaluation. The details about prototype implementation and experimental results are described in Section III. We conclude this paper and discuss our future directions in Section IV.

II. Multiparty Access Control for Google+

A. MPAC+ 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+). Existing OSNs

978-1-4673-0921-9/12/$31.00 ©2012 IEEE

including Google+ do not provide effective mechanism to support collaborative privacy control over shared data. Several access control schemes (e.g., [2], [3]) 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, in addition to the *owner* (the user owning the content in his/her space) of content, other controllers, including the *contributor* (the user publishing the content in someone else's space), *stakeholder* (the user tagged and associated with the content) and *disseminator* (the user sharing 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.

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 co-workers who are with *high* trust levels. Thus, in our multiparty access control model called MPAC+, 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.

We now formally define our MPAC+ model as follows:

- $U = \{u_1, \ldots, u_n\}$ is a set of users of the OSN. Each user has a unique identifier;
- $C = \{c_1, \ldots, c_m\}$ 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 $ct \in CT$:
  $controllers(o : O, ct : CT) = \{u \in U \mid (u, o) \in CO_{ct}\}$;
- $user\_own\_circles : U \rightarrow 2^C$, 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] $\in CUT\}$;
- $user\_extended\_circles : U \rightarrow 2^C$, 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:
  $trust\_level(c : C, u : U) = \{tl \in TL \mid (c, u, tl) \in CUT\}$;

B. MPAC+ Policy Specification

Our policy specification scheme is constructed based on the proposed MPAC+ model. In our model, each controller of a shared resource can specify one or more rules as a policy that can govern 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 1: (**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, we introduce *sensitivity levels* for data specification, which are assigned

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

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 2: (**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 MPAC+ access control rule as follows:

Definition 3: (**MPAC+ Rule**). A MPAC+ 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 1.
- $data$ is represented with a data specification defined in Definition 2; 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. Also, 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. Suppose a controller can leverage five values: 0.00 (*none*), 0.25 (*low*), 0.50 (*medium*), 0.75 (*high*), and 1.00 (*highest*) to represent both sensitivity levels and trust levels, 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:

$r_1 = (Alice, SH, \{< Friends, 0.50 >, < Colleagues, 0.50 >\}, < funny.jpg, 0.75 >, permit)$.

C. MPAC+ Policy Evaluation

In our MPAC+ model, we adopt three steps to evaluate an access request over multiparty access control policies as shown in Figure 1. The first step checks the access request against the policy specified by each controller and yields a decision for the controller. In our MPAC+ 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

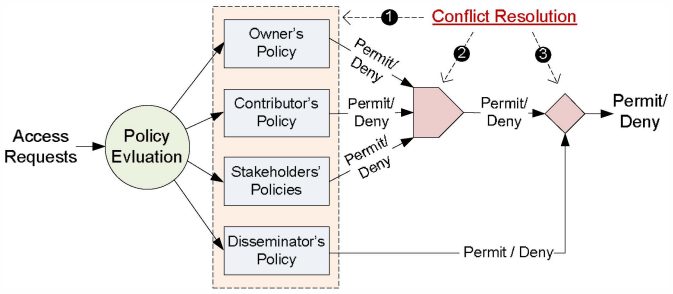


Fig. 1. MPAC+ Policy Evaluation Process.

evaluation mechanism. In the second step, decisions from all controllers in response to the access request are aggregated to make a *collaborative* decision for the access request. Since these controllers may generate different decisions (`permit` and `deny`) for the access request, conflicts may occur. The subsequent sections will address our approach for resolving such conflicts in detail. 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.

The process of conflict resolution is to make a decision to allow or deny the requester's access to the shared data. In general, allowing a requester to access the content may cause *privacy risk*, but denying a requester to access the content may result in *sharing loss*. We adopt a privacy conflict resolution mechanism to balance privacy protection and the users'desire for information sharing through quantitative analysis of *privacy risk* and *sharing loss* [6].

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.

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.

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

where, tl_i denotes the *average* trust level of the accessor i;

function $controllers_d(i)$ returns all denied controllers of an access request i; pc_j denotes the general privacy concern of a denied controller j; and sl_j denotes the sensitivity level of the shared content explicitly chosen by a denied controller j.

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:

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

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

Conflict Resolution: The following equation can be utilized to make the decisions (permitting or denying an access request) for privacy conflict resolution.

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

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

III. Implementation and Evaluation

A. Prototype System Implementation

We implemented a proof-of-concept social network application to demonstrate collaborative management of photos, called *Sigma* (http://apps.facebook.com/sigma_tool). 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 using Facebook users' data to simulate an environment like Google+.

Figure 2 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 3 (c), 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

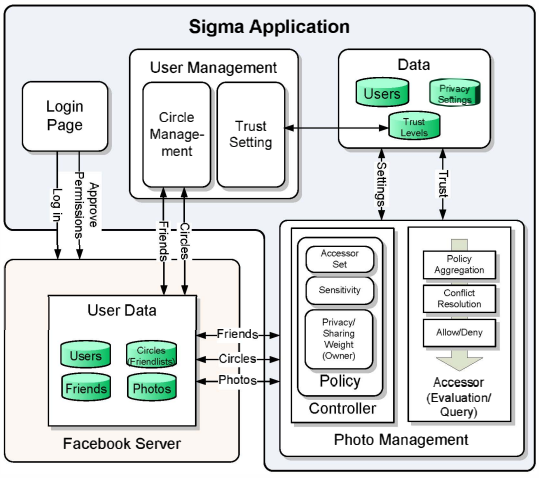


Fig. 2. System Architecture of *Sigma*.

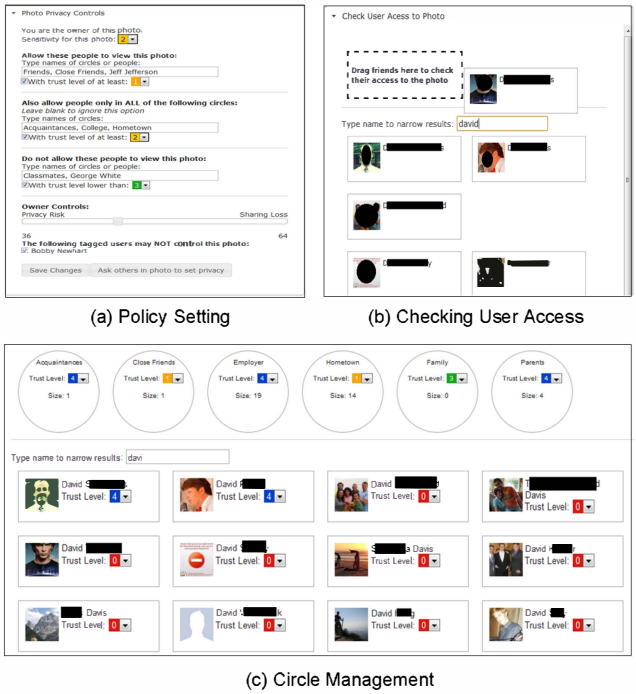
users in that circle in our current implementation. In a real-life implementation, the function of circle trust level would depend on the type of circle. If it is a trust-based circle, trust level may be used as an indication of which users to place in that circle. If it is a group-based circle, it might display an average trust level of all the users.

In the photo management module of *Sigma*, Figure 3 (a) depicts the policy setting. 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 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 3 (b).

B. Prototype System Evaluation

1) User Study: We conducted a user study to test the usability of *Sigma*. [2] 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

[2] http://edu.surveygizmo.com/s3/779289/Sigma-User-Study



(a) Policy Setting (b) Checking User Access

(c) Circle Management

Fig. 3. *Sigma* Interfaces.

3 out of 5 gives a score of 2. Responses from all users are then totaled for comparison.

Prior to using Sigma: Part of the purpose of the user study 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 privacy 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).

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.

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 I. Users ranked the ability to indicate trust almost as important as simplicity, meaning they reacted very positively to this feature of our system.

TABLE I
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

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 II). 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.

TABLE II
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

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 4.

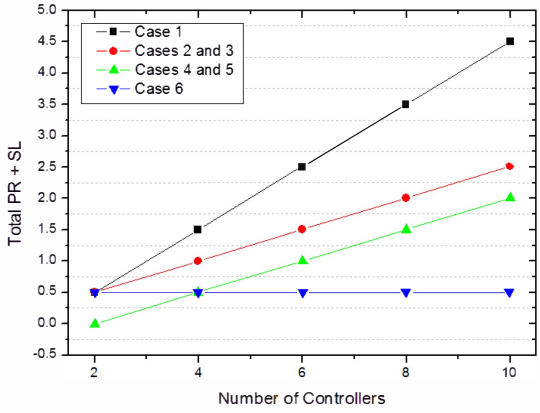


Fig. 4. Privacy Risk and Sharing Loss in Google+ and *Sigma* in Six Cases.

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.

IV. Conclusion and Future Work

In this paper, we have proposed a novel mechanism for collaboratively controlling the shared data in Google+. A multiparty access control model has been formulated. A proof-of-concept implementation of our solution called *Sigma* and the system 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, we would study inference-based techniques [11] for both smarter circle management and automatic configuration of privacy preferences in Google+. Moreover, we plan to conduct model and policy analysis [4], [8] for multiparty access control in OSNs.

Acknowledgments

This work was partially supported by the grants from National Science Foundation (NSF-IIS-0900970 and NSF-CNS-0831360).

References

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

[2] 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.

[3] 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.

[4] H. Hu and G. Ahn. Enabling verification and conformance testing for access control model. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 195–204. ACM, 2008.

[5] 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*, pages 29–43. Springer-Verlag, 2011.

[6] H. Hu, G. 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, pages 103–112. ACM, 2011.

[7] H. Hu, G. Ahn, and J. Jorgensen. Multiparty access control for online social networks: model and mechanisms. *IEEE Transactions on Knowledge and Data Engineering*, pp(99), 2012.

[8] 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.

[9] S. Kairam, M. Brzozowski, D. Huffaker, and E. Chi. Talking in circles: selective sharing in google+. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 1065–1074. ACM, 2012.

[10] 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.

[11] 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.