Policy-by-Example for Online Social Networks

Gorrell P. Cheek, Mohamed Shehab
College of Computing and Informatics
University of North Carolina at Charlotte
Charlotte, NC 28223, USA
{gcheek, mshehab}@uncc.edu

ABSTRACT

We introduce two approaches for improving privacy policy management in online social networks. First, we introduce a mechanism using proven clustering techniques that assists users in grouping their friends for group based policy management approaches. Second, we introduce a policy management approach that leverages a user's memory and opinion of their friends to set policies for other similar friends. We refer to this new approach as Same-As Policy Management. To demonstrate the effectiveness of our policy management improvements, we implemented a prototype Facebook application and conducted an extensive user study. Leveraging proven clustering techniques, we demonstrated a 23% reduction in friend grouping time. In addition, we demonstrated considerable reductions in policy authoring time using Same-As Policy Management over traditional group based policy management approaches. Finally, we presented user perceptions of both improvements, which are very encouraging.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Access Controls; H.5.3 [**Information Interfaces and Presentation**]: Group and Organizational Interfaces

General Terms

Security, Human Factors

Keywords

Policy, Access Control, Grouping, Privacy, Social Network

1. INTRODUCTION

Social networking sites are experiencing tremendous adoption and growth. The internet and online social networks, in particular, are a part of most people's lives. eMarketer[1] reports that in 2011, nearly 150 million US internet users will interface with at least one social networking site per month. eMarketer also reports that in 2011, 90% of internet users ages 18-24 and 82% of internet users ages 25-34 will interact with at least one social networking site per month. This trend is increasing for all age groups. As the young population ages, they will continue to leverage social media in their daily lives. In addition, new generations will come to adopt the internet and online social networks. These technologies have become and will continue to be a vital component of our social fabric which we depend on to communicate, interact and socialize.

Not only are there a tremendous amount of users online, there is also a tremendous amount of user profile data and content online. For example, on Facebook[2], there are over 30 billion pieces of content shared each month. New content is being added every day; an average Facebook user generates over 90 pieces of content each month. This large amount of content coupled with the significant number of users online makes maintaining appropriate levels of privacy very challenging.

There have been numerous studies concerning privacy in the online world [4, 15, 18]. A number of conclusions can be drawn from these studies. First, there are varying levels of privacy controls, depending on the online site. For example, some sites make available user profile data to the internet with no ability to restrict access. While other sites limit user profile viewing to just trusted friends. Other studies introduce the notion of the privacy paradox, the relationship between individual privacy intentions to disclose their personal information and their actual behavior [21]. Individuals voice concerns over the lack of adequate controls around their privacy information while freely providing their personal data. Other research concludes that individuals lack appropriate information to make informed privacy decisions [2]. More over, when there is adequate information, short-term benefits are often opted over long-term privacy. However, contrary to common belief, people are concerned about privacy [1, 9]. But, most are not doing anything about it. This can be attributed to many things, e.g., the lack of privacy controls available to the user, the complexity of using the controls [26] and the burden associated with managing these controls for large sets of users.

We believe that additional tools need to be placed in the hands of the user to aid them in managing their privacy. Our research is focused in two areas. First, we aim to assist users in grouping their large friend sets for privacy policy management purposes. Next, we aim to provide an improved ap-

[1]http://www.eMarketer.com

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

SACMATÕ12, June 20–22, 2012, Newark, New Jersey, USA.

proach for managing access to user profile data and content in online social networks. Our contribution is three-fold:

- We introduce a user assisted friend grouping mechanism that enhances traditional group based policy management approaches. Assisted Friend Grouping leverages proven clustering techniques to aid users in grouping their friends more efficiently. Our approach has demonstrated promising results in assisting users in efficiently grouping and setting expressive policies for their friends. In addition, user perceptions are encouraging.
- We introduce a policy management approach for online social networks that leverages a user's memory and opinion of their friends to set policies for other similar friends, which we refer to as Same-As Policy Management. Using a visual policy editor that takes advantage of friend recognition and minimal task interruptions, Same-As Policy Management demonstrated improved performance and user perceptions over traditional group based policy management approaches.
- We implemented a prototype Facebook application and conducted an extensive user study evaluating our improvements to privacy policy management in online social networks.

The rest of the paper is organized as follows: In Section 2, we provide a brief background of role/group based access control. Section 3 details our two improvements to privacy policy management in online social networks: Assisted Friend Grouping and Same-As Policy Management. Our user study design is described in Section 4 with the results and discussion detailed in Sections 5 and 6, respectively. Finally, we wrap up the paper with related work and conclusions.

2. BACKGROUND

Current social networking platforms offer a simple policy management approach. Security aware users are able to specify policies for their profile objects. For example, my work colleague is restricted from seeing my photos. But, my trusted best friend from school may access all my information. Facebook provides an optional mechanism that allows users to create custom *lists* to organize friends and set privacy restrictions. Similarly, Google+ allows users to create *Circles* of friends, such as family, acquaintances, etc., where the user can apply policies based on these *Circles*. Facebook also recently announced *smart lists* which automatically group friends who live near by or attend the same school. However, managing access for hundreds of *friends* is still a very difficult and burdensome task [17]. In addition, security unaware users typically follow an open and permissive default policy. As a result, the potential for unwanted information leakage is great [23]. We believe that current capabilities to manage access to user profile information on today's social networking platforms are inadequate.

One approach that has been taken to alleviate the burden of managing access permissions for large sets of *friends* is the implementation of a role based access control model (RBAC) [10, 25, 24]. Role based access control provides a level of abstraction with the introduction of a role between the subject and the object permission. A role is a container with a functional meaning, for example, a specific job within an enterprise. Permissions to objects are assigned to roles and subjects are assigned to roles. Role members are granted objective permissions associated with the role(s) in which they belong. See Figure 1. This level of abstraction alleviates the burden of managing large numbers of subject to objective permissions assignments. For the purposes of discussion, we will use the term *group* as to be synonymous with the term *role*, with the understanding that traditionally *roles* have subjects and objects permission assignments and *groups* traditionally only have subject assignments.

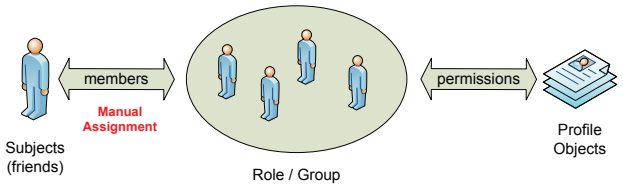


Figure 1: Role Based Access Control

Traditional RBAC can be leveraged within social networks. Often, people's relationships drive privacy decisions. People like to specify groups for their friend relationships, in which they then can set privacy policies [13, 22]. We refer to this approach as group based policy management. However, populating relationship groups can be very time consuming and burdensome to the user [14]. We introduce a group based policy management model that assists users in placing their subjects (or *friends*) into relationship groups. Our approach leverages proven clustering techniques to aid the user in grouping their friends more efficiently. In addition, we provide a mechanism to set friend-level exceptions within group policies. Our model is referred to as the Assisted Friend Grouping Model.

A shortcoming of the group based policy management approach is that the user's attention (mental model) is focused in multiple areas. For example, a user must first focus on the friend's relationship in order to group them appropriately. Next, the user must change focus to the group in order to set the group-level policy. Finally, the user must switch focus back to the friend in order to set any friend-level exceptions for each group policy. We introduce an approach that overcomes this weakness. Our model leverages users' memory and opinion of their friends to set policies for other similar friends. Studies have shown that users perform more efficiently using recognition based approaches that have minimal task interruptions [7, 12]. Using our visual policy editor, a user selects a representative friend (Same-As Example Friend), assigns appropriate object permissions to this friend and then associates other similar friends to the same policy. Our model is called Same-As Policy Management.

3. POLICY-BY-EXAMPLE

Our Policy-By-Example framework is made up of two access control models: Assisted Friend Grouping and Same-As Policy Management. We implemented both models as a prototype Facebook application. The details of which are discussed in the following sections.

3.1 Assisted Friend Grouping

Group based policy management allows users to populate groups based on relationship and assign object permissions

to the groups, refer to Figure 1. Assisted Friend Grouping extends this model in two areas: 1) provides the user with assistance in grouping their friends, and 2) provides the user the ability to set friend-level exceptions within the group policy. See Figure 2.

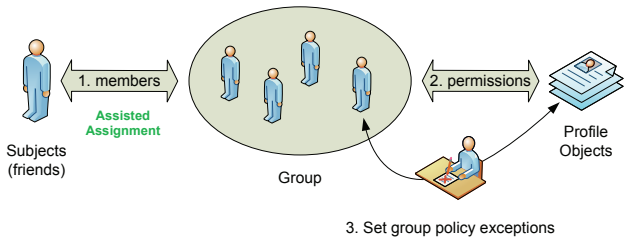


Figure 2: Assisted Friend Grouping Model

For the purposes of our prototype Facebook application, we predefined 10 relationship groups: Family, Close Friends, Graduate School, Under Graduate School, High School, Work, I do not know, Friends of Friend, Community and Other. These groups where carefully selected, in part, from the work of Jones et al. [14]. They postulate that users group their friends, for controlling privacy, based on six criteria: Social Circles, Tie Strength, Temporal Episodes, Geographical Locations, Functional Roles and Organizational Boundaries. Our friend relationship groups were selected to reflect these criteria.

Within our prototype, each friend is presented to the user in the center of a friend grouping page, refer to Figure 3. The user is asked to select, for each friend, the group that best represents their relationship. They can either "drag" the friend to the appropriate relationship group on the page. Or, the user can click the representative relationship group name. To assist users in populating their relationship groups, we leverage the Clasuet Newman Moore (CNM) network clustering algorithm [5]. This clustering algorithm analyzes and detects community structure in networks by optimizing their modularity. Our prototype clusters the user's social network graph creating CNM clusters (or groups) of friends. During friend grouping, we present the friends to the user in CNM group order as recommendations. For example, Bob has 50 friends and clustering his social network graph using CNM produces five clusters. We present to Bob, as recommendations for grouping, all the friends of one CNM group before presenting the friends of each subsequent CNM group. The premise is that CNM groups roughly align with user defined friend populated relationship groups.

By presenting friends in the order they potentially will be grouped, the friend grouping time can be vastly reduced. The user's mental model is focused on roughly one relationship at a time, e.g., work colleagues. The user can quickly ascertain that the stream of friends being presented are all work colleagues and can be placed in the *Work* group. This approach reduces the number of "mental task switches" the user must perform between multiple relationship groups. After all the friends are grouped, the user sets the group policy by setting permissions that allow or deny access to the user's profile objects, e.g., email address, photos, etc. Finally, we provide the user the ability to set friend-level exceptions for each group policy. For example, a group policy may deny access to the user's email address except for group member *Alice*. Most social networking platforms also provide a policy exception setting capability.

Figure 3: Friend Grouping

3.2 Same-As Policy Management

In group based policy management, the user must first group their friends. After which, they must select group permissions (setting the group policy). Finally, friend-level exceptions to the group policy are set. A user's attention (mental model) is focused in multiple areas. Whereas, in Same-As Policy Management, the user's attention is focused on a specific friend. The user leverages their memory and opinion of a friend to set policies for other like friends. In essence, we use a friend recognition approach, with minimal task interruptions, to aid the user in setting policies. A representative friend is selected (Same-As Example Friend), profile object permissions are assigned to this example friend and other similar friends (Same-As Friends) are associated with the same set of object permissions. Figure 4 illustrates our model; the Same-As Example Friend is depicted in front of the user's other similar friends who have been assigned the same set of object permissions.

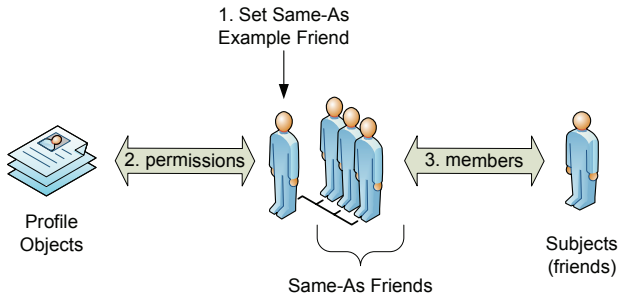


Figure 4: Same-As Policy Management Model

First, the user selects a friend (Same-As Example Friend)

that is representative of a subset of their friend set. The notion is that we all have subsets of friends that have similar levels of trust. The user selects one easy to remember friend from each subset as its respective representative.

Second, using our visual policy editor, the user assigns the appropriate object level permissions for each object within their profile to this Same-As Example Friend. For the purposes of our prototype Facebook application, we presented three profile object categories: *Albums*, *About Me* and *Education and Work*. Within each profile object category, objects of the same family are presented. For example, *About Me* includes Birthday, Status, Current City, email, etc., as indicated in Figure 5. The user can allow or deny access to any object or object category by simply clicking on the object or object category. For example, if the user doesn't want the Same-As Example Friend to have access to a specific photo album, they merely click on that album and the object permission is set to deny. The selected photo album will be grayed out. Or, for example, if the user doesn't want to allow access to any of their education and work information, they click on the object category *Education and Work* and the entire object category will be grayed out, thus effectively setting the permissions to deny for each profile object within that category. Any permutation of permissions is allowed.

Third, after the permissions are set for the Same-As Example Friend, other like or similar friends (Same-As Friends) are assigned to the policy. The visual policy editor presents to the user their friend set, where the user can associate a friend to an already defined Same-As Example Friend. Or, the user can designate a friend as a new Same-As Example Friend, thereby setting a new policy which would be assigned to other similar friends. This process repeats itself for the user's entire friend set.

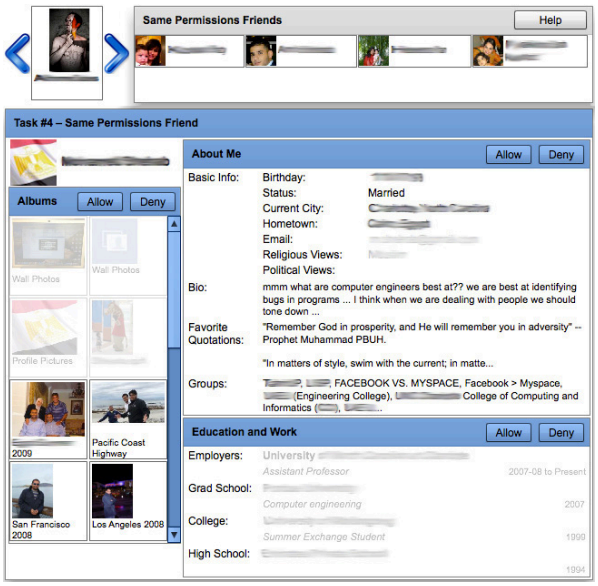


Figure 5: Visual Policy Editor (Blurred for Paper Anonymity Purposes)

3.3 Prototype Architecture

We implemented a prototype of our Assisted Friend Grouping Model and our Same-As Policy Management Model. The prototype is implemented as a Facebook application called PolicyMngr[3]. The application is hosted on our server. The back-end is based on PHP and MySQL. The client-side was implemented using Adobe Flex as a flash application. Upon installing the application, REST like Facebook APIs and Facebook Query Language are used to retrieve the user's profile and social connections. The collected data is transmitted over secure HTTPS based APIs to our server and stored in a MySQL database. The application builds the participant's social graph, which is clustered using the CNM implementation provided by the Flare Toolkit Library[4]. The application implements several additional functionalities, including user grouping, group policy specification, Same-As policy specification and survey tools.

4. USER STUDY

In designing our user study[5], we set out to answer the following research questions:

Q1. Can proven clustering techniques assist users in grouping their friends more efficiently?

Q2. What are users' perceptions of assisted friend grouping techniques?

Q3. Will a policy management approach based on leveraging a user's memory and perception of their friends outperform traditional group based policy management approaches?

Q4. Will users' perceptions of a policy management approach based on leveraging a user's memory and perception of their friends be higher than traditional group based policy management approaches?

4.1 Design

In order to answer these research questions, we built four tasks and two surveys into our Policy-By-Example prototype. The first three tasks and the first survey were designed to evaluate traditional group based policy management and our Assisted Friend Grouping Model. The fourth task and the second survey were designed to evaluate our Same-As Policy Management Model. See Table 1.

Table 1: User Study Tasks

Group Based Policy Management	
Task 1	Group friends
Task 2	Set group policy
Task 3	Review and possibly set friend-level exceptions to group policy
Survey 1	Complete a brief survey for Tasks 1-3
Same-As Policy Management	
Task 4	Set permissions for friends using another friend's permissions as the model/example
Survey 2	Complete a brief survey for Task 4

In the first task (Task 1), the user is instructed to place 50 of their randomly selected friends into the ten predefined

[3]https://apps.facebook.com/policymngr/
[4]http://flare.prefuse.org
[5]Approved IRB Protocol #11-08-01

groups. We divided the user participants into two groups, namely *Not Assisted* and *Assisted*. For the Not Assisted population, the 50 friends were presented to the user for grouping in random order. For the Assisted population, the 50 friends were presented to the user for grouping in CNM group order, as described in Section 3.1. Friends were presented to the user for grouping based on clustering the user's social graph using the CNM algorithm. We measured the grouping time for both populations. After the user placed their friends into groups, they were asked to select access permissions for each group (Task 2). Allow/Deny permissions were selected for each profile object and/or profile object category. Finally in Task 3, the user was asked to review and possibly select friend-level exceptions to the group policy that was set in Task 2.

Upon completion of Tasks 1, 2 and 3, the user was asked to complete the first survey. The initial part of the survey collected basic demographic information summarized in Section 4.2. In the remaining portion of the survey, the user responded to questions designed to capture their perceptions of group based policy management, both the Not Assisted and Assisted friend grouping approaches. The question responses are on a Likert-scale of 1 (Strongly Disagree) to 7 (Strongly Agree). Each question is designed to capture the user's perceptions in the following areas:

Ease of Use: The user needs to be able to manage their policies in an easy, intuitive and effective way such that they have a consistent experience. Complex and laborious policy management mechanisms can lead to ineffective policies.

Readability: Not only does a policy management solution have to be easy to use, it must be decipherable. The core component of any access control mechanism is the policy which governs the access. The policy not only must be available and visible to the user, but it also must be readable. Policies that are complex and difficult to understand are more likely to be misconfigured resulting in unintended consequences, e.g., data leakage.

Flexibility: Policy management mechanisms must be flexible to accommodate the user's needs and intentions. Effective policy management must create a balance between coarse grained and fine grained access control. Traditionally, coarse grained access control provides few options to the end user. On the other hand, fine grained access control, although extremely flexible in that it provides lots of options and capabilities, is traditionally overwhelming and complex. A balance between too little flexibility and an overly burdensome policy management mechanism is needed.

The fourth task was designed to evaluate our Same-As Policy Management Model, as described in Section 3.2. The user was instructed, for a subset of their friends (50 randomly chosen ones), to select a Same-As Example Friend, set appropriate profile object permissions for this example friend and assign the policy to appropriate like or similar friends. This step was repeated as necessary, i.e., for as many unique policies the user would like to assign for their friend set. We measured the total time to complete Task 4. After completing Task 4, the user completed a second survey identical to the first survey excluding the demographic questions.

4.2 Participants

We recruited our user study participants from the student and staff population of the university community and from Amazon Mechanical Turk[6]. Amazon Mechanical Turk is a crowd sourcing marketplace that pairs *Requesters* of work and *Workers*. Requesters formulate work into Human Intelligent Tasks (HIT) which are individual tasks that workers complete. We set up our prototype Facebook application as a HIT. This included all four tasks and the two surveys, as described in Section 4.1. To better control the quality of the recruited participants, we mandated that each worker have a 95% HIT approval rating, or better. A HIT took approximately 10-15 minutes to complete, for which each worker was paid a fee of $1.50. A total of 101 users successfully completed the user study. We used the total time spent to complete the study as a measure to remove 5% of the outlying users who had an absolute Z-Score value greater than three.

That left 96 participants in our study, 77 male and 19 female. Most of our user participants were young, fairly well educated and active Facebook users. 67% were between the ages of 18 to 25. 74% had between two and four years of college. Almost 83% used Facebook daily. In addition, as part of the demographics portion of our survey, we collected Westin privacy sentiment information summarized below with definitions of *Unconcerned*, *Pragmatist* and *Fundamentalist* provided by [16]:

Unconcerned Users: 13.5% of our user study population. *This group does not know what the "privacy fuss" is all about, supports the benefits of most organizational programs over warnings about privacy abuse, has little problem with supplying their personal information to government authorities or businesses, and sees no need for creating another government bureaucracy (a Federal Big Brother) to protect someone's privacy.*

Pragmatists: 62.5% of our user study population. *This group weighs the value to them and society of various business or government programs calling for personal information, examines the relevance and social propriety of the information sought, wants to know the potential risks to privacy or security of their information, looks to see whether fair information practices are being widely enough observed, and then decides whether they will agree or disagree with specific information activities - with their trust in the particular industry or company involved being a critical decisional factor.*

Fundamentalists: 24% of our user study population. *This group sees privacy as an especially high value, rejects the claims of many organizations to need or be entitled to get personal information for their business or governmental programs, thinks more individuals should simply refuse to give out information they are asked for, and favors enactment of strong federal and state laws to secure privacy rights and control organizational discretion.*

5. STUDY RESULTS

The next two subsections detail our user study results for the Assisted Friend Grouping Model and Same-As Policy Management.

5.1 Assisted Friend Grouping

In evaluating our Assisted Friend Grouping Model, we set out to show that CNM will aid in grouping users' friends more efficiently for group based policy management approaches.

[6]https://www.mturk.com/

Our hypothesis is that CNM clusters roughly align with user defined friend relationship groups.In the example illustrated in Figure 6, CNM partitions the user's social graph into distinct clusters, as depicted by the large circles. The user also categorizes their friends into user defined relationship groups, i.e., Family, Graduate School, etc. Figure 6 illustrates that there is overlap and agreement between the CNM clusters and the user defined relationship groups. We leverage this alignment by presenting friends to the user for grouping based on cluster/relationship order. By presenting friends in this manner, the user's mental model is focused on one relationship at a time. This approach results in fewer "mental task switches" between multiple relationship groups and thus improved friend grouping times.

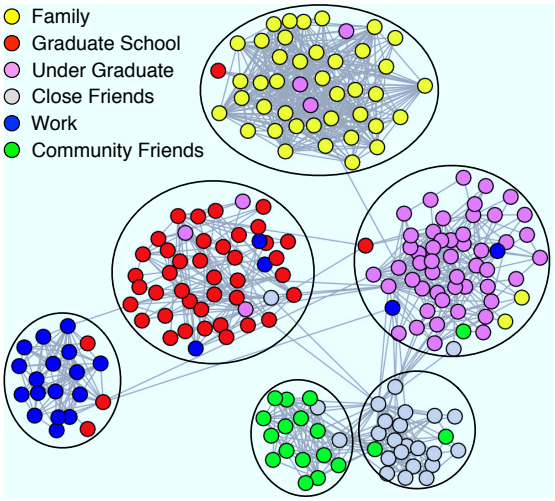


Figure 6: Example Cluster/Group Alignment

We used the Adjusted Rand Index to measure the agreement between CNM clusters and user defined relationship groups [11]. The Adjusted Rand Index compares the predicated labels (CNM clusters) with the actual labels (user defined relationship groups) and produces an index between 0 and 1, where 0 indicates no overlap and 1 is complete agreement or overlap. The Adjusted Rand Index, in general form, can be described as $\frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex}$. We clustered users' social graphs who were Not Assisted in grouping their friends, i.e., we presented their friend set for grouping in random order. We compared the clusters generated by CNM and the populated groups defined by the user. We found, that on average, users populated 6.4 relationship groups. Overall, our results showed an average Adjusted Rand Index of 0.653. This demonstrates that there is overlap and a level of alignment between CNM clusters and user defined relationship groups. In looking just at Unconcerned Users, we saw a higher level of alignment (Adjusted Rand Index = 0.784).

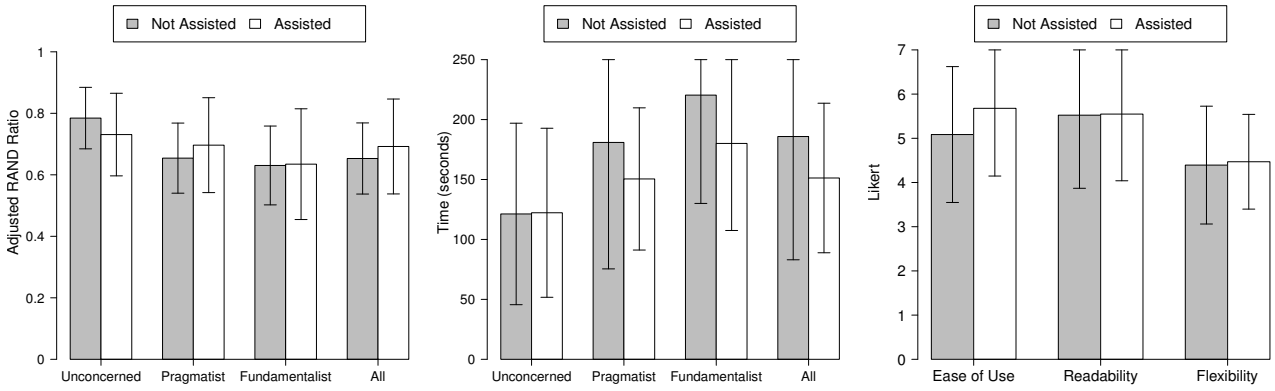
We also wanted to determine if presenting friends in CNM group order would influence the user in how they grouped their friends. We compared the Assisted friend grouping population with those that were Not Assisted. Using a Welch Two-Sample T-Test, we found no statistical significance between the two populations ($p = 0.224$). Refer to the Adjusted Rand Index section of Table 2 and Figure 7(a), where error bars show one standard deviation above and below the mean. Our Assisted Friend Grouping Model does not bias the user, i.e., the user would produce the same groups and populate those groups with the same friends either using our Assisted Friend Grouping approach or not.

Table 2: Not Assisted vs. Assisted Friend Grouping

Measure	Not Assisted (μ, σ)	Assisted (μ, σ)	p-value
Adjusted Rand Index			
Unconcerned	(0.784, 0.10)	(0.731, 0.13)	0.383
Pragmatist	(0.654, 0.12)	(0.696, 0.15)	0.183
Fundamentalist	(0.631, 0.13)	(0.634, 0.18)	0.761
All	(0.653, 0.12)	(0.692, 0.15)	0.224
Grouping Time (seconds)			
Unconcerned	(124.1, 75.7)	(125.4, 70.5)	0.85
Pragmatist	(180.9, 105.48)	(150.4, 59.31)	0.012
Fundamentalist	(220.4, 90.36)	(171.8, 69.82)	0.026
All	(185.8, 102.8)	(150.6, 61.84)	0.038
User Perceptions (7 point Likert-scale)			
Ease of Use	(5.08, 1.54)	(5.70, 1.53)	0.041
Readability	(5.52, 1.65)	(5.57, 1.51)	0.861
Flexibility	(4.39, 1.33)	(4.49, 1.07)	0.681

Next, we set out to measure the time it took a user to populate their relationship groups. We measured the time it took a user to group 50 of their friends presented in random order (Not Assisted). We compared that with the time it took the same user to group 50 of their friends presented in CNM group order (Assisted), as described in Section 3.1. For Unconcerned Users, there was no statistical significance between Not Assisted and Assisted ($p = 0.85$). However, we did see statistical significance between the other categories of users: Pragmatists ($p = 0.012$), Fundamentalists ($p = 0.026$) and the population as a whole ($p = 0.038$). Overall, using CNM, we saw a 23% reduction in time that it took a user to group 50 of their friends, 150.6 seconds (Assisted) versus 185.8 seconds (Not Assisted). Refer to the Grouping Time section of Table 2 and Figure 7(b). One factor for this reduction in time is that the user's mental model is focused on one relationship group at a time, which enables the user to quickly group most family members, for example, before grouping the next set of friends. Fewer "mental task switches" between relationship groups are required thus reducing the overall friend grouping time. It is also interesting to note, although not entirely surprising, that Fundamentalists took longer, on average, to group their friends than Pragmatists and Unconcerned Users. One possible reason that Fundamentalists took more time may be because they apply more scrutiny as they group their friends.

We also measured users' perceptions of the Not Assisted and Assisted friend grouping approaches, as described in Section 4.1. A T-Test was used to compared the Not Assisted and Assisted populations. We found no statistical significance in the areas of Readability ($p = 0.861$) and Flexibility ($p = 0.681$). This would be expected because the policy management approaches for Not Assisted and Assisted are visibly the same with the only difference being the order in which friends are presented. However, we do see statistical significance in the area of Ease of Use ($p = 0.041$), Not Assisted averaged 5.08 and Assisted averaged 5.70 on a 7 point Likert-scale. Refer to the User Perceptions section of



(a) Adjusted Rand Index (b) Grouping Time (c) User Perceptions

Figure 7: Not Assisted vs. Assisted Friend Grouping

Table 2 and Figure 7(c). Users found grouping their friends leveraging CNM easier than not having the assistance of CNM.

5.2 Same-As Policy Management

We compared the policy authoring times between Group Based Policy Management (hereafter referred to as Group Based) and Same-As Policy Management (hereafter referred to Same-As). Our results are summarized in the Policy Authoring Time section of Table 3 and illustrated in Figure 8(a). In analyzing these results, we found that there is statistical significance across all user categories, i.e., Unconcerned Users ($p = 0.036$), Pragmatists ($p < 0.001$) and Fundamentalists ($p =< 0.001$). Overall, Same-As outperformed Group Based in policy authoring time. Across the board, we observed more than a two-fold decrease in the amount of time it took a user to author their policy. One factor attributing to this reduction is the steps involved in authoring a policy. Group Based approaches have three distinct steps: 1) group friends, 2) set group policy and 3) assign friend-level exceptions to the group policy. Using this approach, the user first focuses on the friend's relationship in order to group them appropriately. Next, the user switches their attention to the group in order to set the group policy. Finally, the user switches their attention back to the friend in order to set any friend-level exceptions to the group policy. Whereas, using our Same-As approach and visual policy editor, the user simply leverages their memory and opinion of a friend to set policies for other similar friends. As a result, users can author policies in less time and thus ease the burden associated with managing their online privacy settings.

Not only are users able to set their policies more rapidly using Same-As, they are also setting more conservative policies, policies that are less permissive. We examined the *openness* of each user's policy, where Policy Openness is defined as:

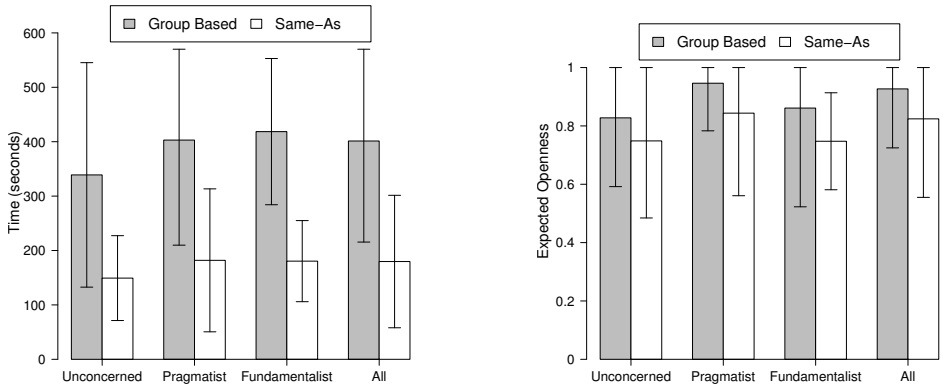
DEFINITION 1. *(Policy Openness) The probability of a user permitting a friend access to a specific profile object.* $O(u,o) = \frac{|Allow(f,o)|}{|F_u|}$, *where* $Allow(f,o) \subseteq F_u$ *is the set of friends of user* u *who are allowed access to profile object* o *and* F_u *is the friend set of* u.

Table 3: Group Based vs. Same-As

Measure	Group Based (μ, σ)	Same-As (μ, σ)	p-value
Policy Authoring Time (seconds)			
Unconcerned	(338.9, 206.3)	(149.2, 77.9)	0.036
Pragmatist	(402.9, 193.1)	(181.9, 131.3)	< 0.001
Fundamentalist	(418.5, 134.4)	(180.4, 74.5)	< 0.001
All	(401.2, 185.8)	(179.7, 121.7)	< 0.001
Policy Openness (see Definition 1)			
Unconcerned	(0.827, 0.235)	(0.748, 0.264)	0.596
Pragmatist	(0.946, 0.163)	(0.843, 0.283)	0.006
Fundamentalist	(0.869, 0.338)	(0.751, 0.166)	0.022
All	(0.927, 0.202)	(0.823, 0.269)	0.002

We measured Policy Openness relative to a user's profile object (i.e., email address) and found, for Unconcerned Users, no statistical significance between Group Based and Same-As ($p = 0.596$). Unconcerned Users have "little problem with supplying their personal information" to others in either approach. However, we do see statistical significance between Group Based and Same-As for Pragmatists ($p = 0.006$), Fundamentalists ($p = 0.022$) and for the population as a whole ($p = 0.002$). Our findings are summarized in the Policy Openness section of Table 3 and Figure 8(b). Using Group Based, users associate the policy with a group. Whereas, using Same-As, users associate the policy with a friend and in doing so have the friend in the forefront of their mind. This allows users to be more selective and careful in assigning permissions. Users are thinking of people, not groups. In addition, as would be expected, our results show that Fundamentalists write more conservative policies than Pragmatists and Unconcerned Users.

Overall, users found Same-As easier to use than Group Based, 5.97 versus 5.38 on a 7 point Likert-scale, where 7 is Strongly Agree. We found statistical significance in our comparison ($p = 0.007$). Refer to Ease of Use section of Table 4 and Figure 9(a). Using Same-As over Group Based, we observed statistical significance and improved Ease of Use ratings for Unconcerned Users ($p = 0.045$) and Pragmatists ($p = 0.008$). We attribute the improved ratings to reasons



(a) Policy Authoring Time

(b) Policy Openness

Figure 8: Group Based vs. Same-As Policy Management

similar to what was discussed with regard to the reduction in policy authoring time: reduced number of steps for authoring policies, our visual policy editor and consistent focus with limited memory interruption. However, from an Ease of Use perspective, there was no statistical significance for Fundamentalists ($p = 0.604$). One possible reason is that fundamentalists are very concerned about privacy and may consider privacy "hard" to attain regardless of the approach. Also, it is interesting to note that Unconcerned Users averaged Ease of Use ratings higher than Pragmatists and Fundamentalists.Unconcerned Users don't necessarily care much about privacy and appreciate mechanisms that are easier. Fundamentalists find privacy to be "hard" regardless of approach and Pragmatists fall somewhere in the middle.

Table 4: Gp Based vs. Same-As – User Perceptions

Measure	**Group Based** (μ, σ)	**Same-As** (μ, σ)	**p-value**
Ease of Use (7 point Likert-scale)			
Unconcerned	(6.32, 0.47)	(6.88, 0.42)	0.045
Pragmatist	(5.37, 1.49)	(6.01, 1.45)	0.008
Fundamentalist	(4.86, 1.99)	(5.26, 2.07)	0.604
All	(5.38, 1.55)	(5.97, 1.54)	0.007
Readability (7 point Likert-scale)			
Unconcerned	(5.01, 0.48)	(6.80, 0.18)	< 0.001
Pragmatist	(4.49, 1.17)	(5.73, 1.34)	< 0.001
Fundamentalist	(3.81, 1.45)	(5.02, 1.82)	0.035
All	(4.44, 1.21)	(5.71, 1.42)	< 0.001
Flexibility (7 point Likert-scale)			
Unconcerned	(6.53, 0.48)	(6.70, 0.33)	0.505
Pragmatist	(5.55, 1.55)	(5.74, 1.44)	0.437
Fundamentalist	(4.99, 1.84)	(5.34, 1.99)	0.614
All	(5.55, 1.58)	(5.76, 1.51)	0.336

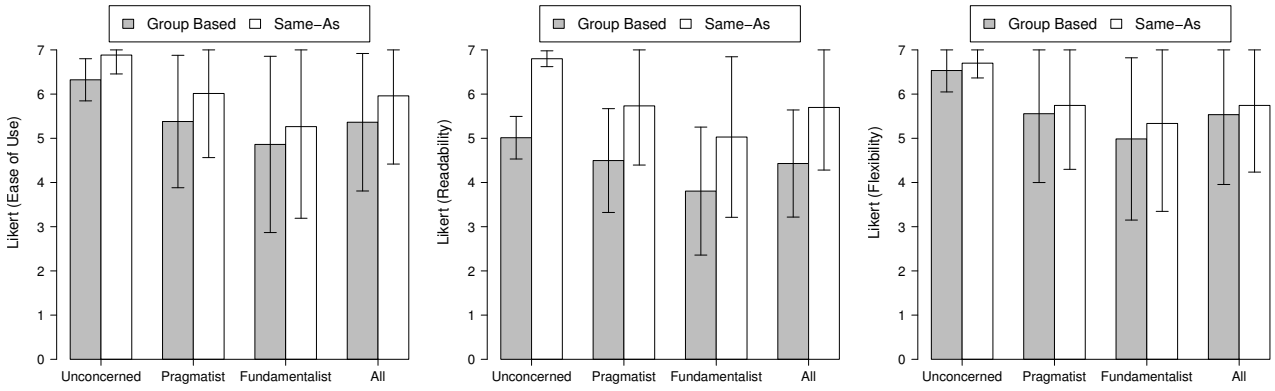
Users found Same-As to be substantially more readable than Group Based. There is statistical significance across all user categories. Refer to the Readability section of Table 4 and Figure 9(b). We attribute these high ratings to the simplicity of the Same-As approach. Users could easily understand who had access to what profile object. Users found the organization of the information on the screen to be decipherable and ease to read. Using Same-As and leveraging our visual policy editor, a user need only to recall their opinions of their friends in order to set access control policies. This was accomplished all on one screen. Whereas, the Group Based approach was more complex with multiple steps and screens.

In evaluating Flexibility, on average, users gave relatively high ratings to both Group Based and Same-As, 5.55 for Group Based and 5.76 for Same-As. However, we found no statistical significance between the two populations ($p = 0.336$). Refer to the Flexibility section of Table 4 and Figure 9(c). In access control terms, both Group Based and Same-As have the same expressive power. That is, users can compose policies of the same granularity with either Group Based or Same-As. Group Based allows finer grained policies with the inclusion of friend-level exceptions to group policies. Same-As inherently has this capability.

6. DISCUSSION

Complex and laborious policy management mechanisms can lead to ineffective policies and compromises of information. Group based policy management is an improvement which provides a level of abstraction to the user (i.e., group) that allows them to manage permissions of large friend sets easier. However, this approach has some limitations, one being the ability to set fine grained access control policies. Introducing the capability to set friend-level exceptions to group policies overcomes this limitation. By doing so, users have the ability to set more expressive access control policies. Another shortcoming of group based policy management approaches is the burden associated with populating relationship groups for large friends sets. Our Assisted friend grouping model alleviates this burden by reducing the amount of time it takes to populate friend groups. User perceptions of our approach are encouraging. Providing tools in the hands of the user, which assist them in managing access to their profile objects, translates into more effective privacy management.

Same-As Policy Management further improves upon group



(a) Ease of Use (b) Readability (c) Flexibility

Figure 9: Group Based vs. Same-As Policy Management – User Perceptions

based policy management. It provides the same level of expressive power for setting fine grained policies. But, doing it in a way that is easier for the user to manage and intuitively easier to comprehend. Using our visual policy editor, users can compose readable policies that are not complex and difficult to understand. In addition, users can compose these policies in about half the time it takes traditional group based policy management approaches. Policy management becomes less of a laborious and tedious task and results in more properly configured and maintained policies, which leads to improved privacy. In addition, Same-As Policy Management results in more conservative policies, which ultimately provide better levels of protection. Same-As Policy Management keeps users more informed, improves the adoption and accuracy of access control policies and, ultimately, improves user security.

There are areas of opportunity with our research. For Assisted Friend Grouping, our prototype Facebook application cannot accommodate friends being placed into more than one relationship group. Currently, our approach recommends a friend be placed in the "best" group. Clearly, there are examples where we would expect a friend to be in multiple groups, e.g., Alice, my sister (*Family Group*), went to the same college (*Undergraduate School Group*) as I did. This is a limitation of our implementation and an area for further research.

Our user study participants were comprised of students and staff that we recruited from the university community and from *Workers* from Amazon Mechanical Turk, as described in Section 4.2. By leveraging a crowd sourcing marketplace, like Amazon Mechanical Turk, there is the possible element of a self-selection bias. Also, we presented to our participants, the Group Based Policy Management tasks followed by the Same-As Policy Management tasks. An improved approach would have been to present the tasks in random order.

7. RELATED WORK

Yuksel et al. [27] propose an approach to managing privacy in online social networks that is based on the grouping of friends, with the assumption that friends share the same information with other group members. They use standard clustering techniques, as we do. In addition, they survey the users by asking them questions that would reveal their willingness to share information with others in their social network. Yuksel et al. take this survey data to further refine their grouping approach. This is an interesting approach. However, they provide little empirical data that would demonstrate its feasibility and effectiveness.

Jones et al. [14] investigate users' rationales for grouping friends, for privacy management purposes, within online social networks. They identify six criteria, which we leverage in choosing our predefined relationship groups. In addition, they evaluate the similarity of these criteria to the output of standard clustering techniques of users' friends. Their work supports our notion that standard clustering techniques can assist users in placing friends into groups analogous with privacy intentions. However, these mechanisms are not 100% accurate. We provide a mechanism that allows users to set exceptions to the grouping output of the automated clustering. By doing so, we can set more expressive policies based on users' intentions.

Mazzia et al. [20] introduce a policy visualization tool which displays privacy settings for user specific subgroups of friends within social networks. Besmer et al. [3] analyze the impacts of community information on access control policy decisions within social networks. Lipford [19] et al. compare two different approaches for representing social network privacy policies. They conclude that there are few differences in user performance. However, each has its strengths over the other. Many other studies have shown the benefits of recognition based approaches in aiding in memory recall [7, 8] and the ill effects of work/task interruption [12, 6]. Same-As Policy Management leverages concentrated memory recognition of friends using a visual policy editor to manage privacy in online social networks.

8. CONCLUSION

In this paper, we introduced two approaches to improving privacy policy management in online social networks. First, we presented an approach, leveraging proven clustering techniques, that assists users in grouping their friends for policy

management purposes. Our approach demonstrated reduced grouping times and improvements in ease of use over traditional group based policy management approaches. Second, we introduced Same-As Policy Management, which leverages a user's memory and opinion of their friends to set policies for other similar friends. Our visual policy editor uses friend recognition and minimal task interruption to obtain substantial reductions in policy authoring times. In addition, Same-As Policy Management was positively perceived by users over traditional group based policy management approaches.

9. ACKNOWLEDGEMENTS

This research was partially supported by grants from the National Science Foundation (NSF-CNS-0831360, NSF-CNS-1117411) and a Google Research Award.

10. REFERENCES

[1] A. Acquisti and R. Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *Privacy Enhancing Technologies*, pages 36–58, 2006.

[2] A. Acquisti and J. Grossklags. Privacy and rationality in individual decision making. *IEEE Security and Privacy*, 3(1):26–33, 2005.

[3] A. Besmer, J. Watson, and H. R. Lipford. The impact of social navigation on privacy policy configuration. In *SOUPS*, 2010.

[4] J. Bonneau and S. Preibusch. The privacy jungle: On the market for data protection in social networks. In *The Eighth Workshop on the Economics of Information Security (WEIS 2009)*, 2009.

[5] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, pages 1– 6, 2004.

[6] E. Cutrell, M. Czerwinski, and E. Horvitz. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. pages 263–269. IOS Press, 2001.

[7] R. Dhamija and A. Perrig. Deja vu: A user study using images for authentication. In *Proceedings of the 9th conference on USENIX Security Symposium - Volume 9*, pages 4–4, Berkeley, CA, USA, 2000. USENIX Association.

[8] P. Dunphy, A. P. Heiner, and N. Asokan. A closer look at recognition-based graphical passwords on mobile devices. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, page 1. ACM, 2010.

[9] C. Dwyer, S. R. Hiltz, and K. Passerini. Trust and privacy concern within social networking sites: A comparison of facebook and myspace. In *Proceedings of the Thirteenth Americas Conference on Information Systems (AMCIS 2007)*, 2007. Paper 339.

[10] D. Ferraiolo and R. Kuhn. Role-based access control. In *In 15th NIST-NCSC National Computer Security Conference*, pages 554–563, 1992.

[11] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[12] S. T. Iqbal and B. P. Bailey. Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. In *CHI '05 extended abstracts on Human factors in computing systems*, CHI EA '05, pages 1489–1492, New York, NY, USA, 2005. ACM.

[13] Q. Jones, S. A. Grandhi, S. Whittaker, K. Chivakula, and L. Terveen. Putting systems into place: a qualitative study of design requirements for location-aware community systems. In *In Proceedings of CSCW*, pages 202–211. ACM, 2004.

[14] S. Jones and E. O'Neill. Feasibility of structural network clustering for group-based privacy control in social networks. In *SOUPS*, 2010.

[15] H. Krasnova, O. Günther, S. Spiekermann, and K. Koroleva. Privacy concerns and identity in online social networks. *Identity in the Information Society*, 2:39–63, 2009.

[16] P. Kumaraguru and L. F. Cranor. Privacy indexes: A survey of westin's studies. *ISRI Tech. Report*, 2005.

[17] S. Lederer, J. I. Hong, A. K. Dey, and J. A. Landay. Personal privacy through understanding and action: five pitfalls for designers. *Personal and Ubiquitous Computing*, 8(6):440–454, 2004.

[18] K. Lewis, J. Kaufman, and N. Christakis. The taste for privacy: An analysis of college student privacy settings in an online social network. *Journal of Computer-Mediated Communication*, 14(1), 2008.

[19] H. R. Lipford, J. Watson, M. Whitney, K. Froiland, and R. W. Reeder. Visual vs. compact: a comparison of privacy policy interfaces. In *CHI*, 2010.

[20] A. Mazzia, K. LeFevre, and E. Adar. The PViz Comprehension Tool for Social Network Privacy Settings. Technical Report CSE-TR-570-11, University of Michigan, April 2011.

[21] P. A. Norberg, D. R. Horne, and D. A. Horne. The Privacy Paradox: Personal Information Disclosure Intentions versus Behaviors. *Journal of Consumer Affairs*, 2007.

[22] J. S. Olson, J. Grudin, and E. Horvitz. A study of preferences for sharing and privacy. In *CHI Extended Abstracts*, pages 1985–1988, 2005.

[23] PCWorld. Google buzz criticized for disclosing gmail contacts. `http://www.pcworld.com/businesscenter/article/189081`, February 2010.

[24] R. Sandhu, D. Ferraiolo, and R. Kuhn. The nist model for role-based access control: Towards a unified standard. In *In Proceedings of the fifth ACM workshop on Role-based access control*, pages 47–63, 2000.

[25] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

[26] K. Strater and H. R. 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*, BCS-HCI '08, pages 111–119, Swinton, UK, UK, 2008. British Computer Society.

[27] A. S. Yuksel, M. E. Yuksel, and A. H. Zaim. An approach for protecting privacy on social networks. In *Proceedings of 5th International Conference on Systems and Networks Communications*, Washington, DC, USA, 2010. IEEE Computer Society.