Human Effects of Enhanced Privacy Management Models

Gorrell P. Cheek, *Member*, *IEEE*, and Mohamed Shehab, *Member*, *IEEE*

Abstract—We enhance existing and introduce new social network privacy management models and we measure their human effects. First, we introduce a mechanism using proven clustering techniques that assists users in grouping their friends for traditional group-based policy management approaches. We found measurable agreement between clusters and user-defined relationship groups. Second, we introduce a new privacy management model that leverages users' memory and opinion of their friends (called example friends) to set policies for other similar friends. Finally, we explore different techniques that aid users in selecting example friends. We found that by associating policy temples with example friends (versus group labels), users author policies more efficiently and have improved perceptions over traditional group-based policy management approaches. In addition, our results show that privacy management models can be further enhanced by utilizing user privacy sentiment for mass customization. By detecting user privacy sentiment (i.e., an unconcerned user, a pragmatist or a fundamentalist), privacy management models can be automatically tailored specific to the privacy sentiment and needs of the user.

Index Terms—Policy, human factors, privacy, access control, 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.com reports that in 2011, nearly 150 million US Internet users will interface with at least one social networking site per month. eMarketer.com also reports that in 2011, 90 percent of Internet users ages 18-24 and 82 percent 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, 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 [5], [23], [26]. 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 [31]. 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 [3]. Moreover, 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 [2], [13]. But managing ones privacy can be challenging. This can be attributed to many things, for example, the lack of privacy controls available to the user, the complexity of using the controls [36], and the burden associated with managing these controls for large sets of users.

We enhance existing and introduce new privacy management models for online social networks. In addition, we measure the human effects of our improvements. We introduce three new improvements to privacy management models:

1. *Assisted Friend Grouping*—an incremental improvement to traditional group-based policy management.
2. *Same-As Policy Management*—a new paradigm improvement over traditional group-based policy management.
3. *Example Friend Selection*—an incremental improvement to Same-As Policy Management.

We leverage traditional group-based policy management as our baseline and progressively improve upon this privacy management model. With each new enhancement,

• The authors are with the Department of Software and Information Systems, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223. E-mail: {gcheek, mshehab}@uncc.edu.

Manuscript received 30 Oct. 2012; revised 6 May 2013; accepted 3 Aug. 2013; published online 20 Aug. 2013.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-2012-10-0267.
Digital Object Identifier no. 10.1109/TDSC.2013.34.

we measure their human effects including cluster/user-defined relationship group alignment, user privacy sentiment, efficiencies and user perceptions.

Our contributions are as follows:

- 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 effectively and efficiently. We found measurable agreement between clusters and user-defined relationship groups. In addition, user perceptions of our improvements are encouraging.
- We introduce a new privacy management model that is an improvement over traditional group-based policy management approaches. Our new paradigm 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*. Users associate the policy with an example friend and in doing so have this 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. 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 further enhance Same-As Policy Management by introducing *Example Friend Selection*—two techniques for aiding users in selecting their example friends that are used in developing policy templates. Both techniques reduced policy authoring times and were positively perceived by users.
- We detect user privacy sentiment that can be leveraged to further enhance privacy management models. For example, Unconcerned Users who author more open policies may leverage a less flexible coarse-grained privacy management approach. Whereas a Fundamentalist, who authors more conservative policies, will find a fine-grained approach better suited for meeting their privacy needs. Privacy management models can be further refined and enhanced by detecting and leveraging user privacy sentiment in managing access to user privacy information.

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 improvements to privacy management models. Our user study design is described in Section 4 with the results/human effects and discussion detailed in Sections 5 and 6, respectively. Finally, we wrap up the paper with related work, conclusions, and future work.

2 Background

Many 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, and so on, where the user can apply policies based on these *Circles*. Facebook also has *smart lists* that automatically group friends who live nearby or attend the same school. However, managing access for hundreds of *friends* is still a very difficult and burdensome task [25]. In addition, security unaware users typically follow an open and permissive default policy. As a result, the potential for unwanted information leakage is great [1].

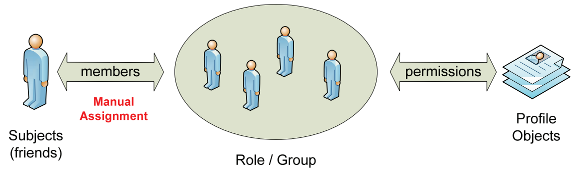


Fig. 1. Role-based access control.

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) [15], [34], [35]. 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 object permissions associated with the role(s) in which they belong. See Fig. 1. This level of abstraction alleviates the burden of managing large numbers of subjects to object permissions assignments. For the purposes of discussion, we will use the term *group* to be synonymous with the term *role*, with the understanding that traditionally *roles* have subject to object permissions assignments and *groups* traditionally only have subject assignments.

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 [21], [32]. We refer to this approach as a group-based policy management. However, populating relationship groups can be very time consuming and burdensome to the user [22]. We enhance traditional group-based policy management by introducing a mechanism that assists users in placing their subjects (or *friends*) into relationship groups. Our approach leverages proven clustering techniques, which have measurable agreement with user-defined relationship groups, to aid users in grouping their friends more efficiently. Our model is referred to as *Assisted Friend Grouping*.

A shortcoming of the group-based policy management approach is that the user's attention is focused in multiple areas. For example, a user must first focus on the friend's relationship to group them appropriately. Next, the user must change focus to the group to set the group-level policy. Finally, the user must switch focus back to the friend to set

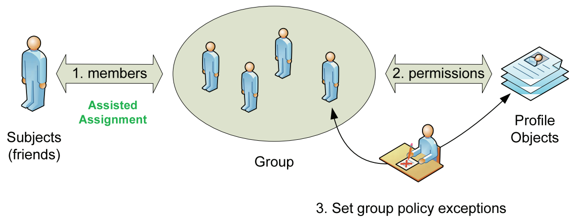


Fig. 2. Assisted Friend Grouping model.

any friend-level exceptions for each group policy. We introduce a new privacy management paradigm that overcomes this weakness. Our model leverages a user's 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 [11], [20]. 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*. We further enhance Same-As Policy Management by introducing two techniques for selecting representative friends (same-as example friends) used in the development of policies. Our model is called *Example Friend Selection*.

3 Enhanced Privacy Management Models

We enhance existing and introduce new social network privacy management models, in addition to measuring the human effects of these models. First, we improve upon traditional group-based policy management with Assisted Friend Grouping. Next, we introduce a new approach for privacy management called Same-As Policy Management. We further improve upon Same-As Policy Management by introducing techniques for selecting friends used in developing policies, called Example Friend Selection. The details of which are discussed in the following sections.

3.1 Group-Based Policy Management with Assisted Friend Grouping

Group-based policy management allows users to populate groups based on relationship and assign object permissions to the groups, refer to Fig. 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 Fig. 2.

For the purposes of our prototype Facebook application, we predefined 10 relationship groups: family, close friends, graduate school, under graduate school, high school, work, acquaintances, friends of friend, community, and other. These groups were carefully selected, in part, from the work of Jones and O'Neil [22]. 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 Fig. 3.

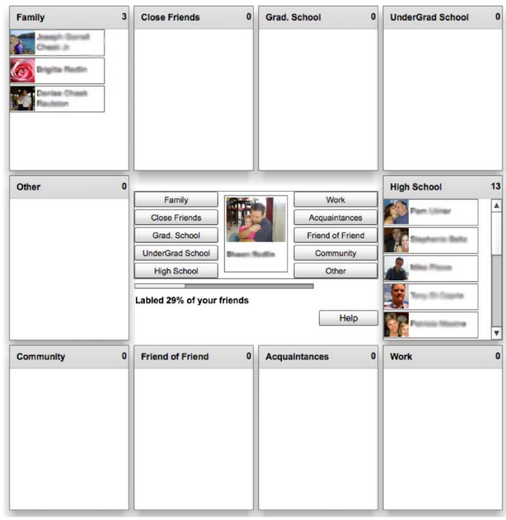


Fig. 3. Friend grouping *(blurred for anonymity)*.

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 [9]. This clustering algorithm analyzes and detects community structure in networks by optimizing their modularity [30]. Modularity is a metric that describes the quality of a specific proposed division of a network into communities. 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 user's mental model is focused on roughly one relationship at a time, for example, 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, for example, e-mail address, photos, and so on.

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. Users leverage their memory and

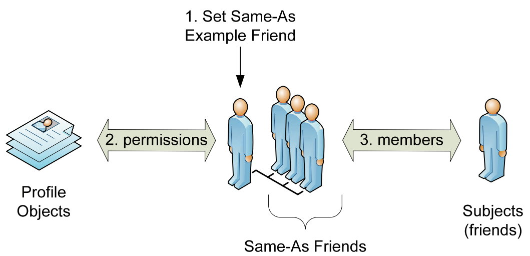


Fig. 4. Same-As Policy Management model.

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. Fig. 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.

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 appropriate object level permissions for each object for 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, and so on, as indicated in Fig. 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 does not want the same-as example friend to have access to their college information, they merely click on College, and the object permission is set to deny and the object will be grayed out. Or, for example, if the user does not want to

Fig. 5. Visual policy editor *(blurred for anonymity)*.

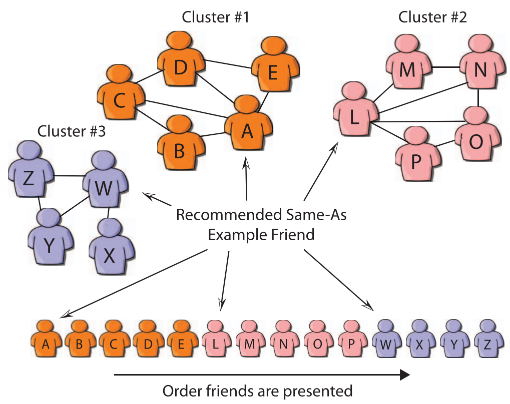


Fig. 6. CNM Order.

allow access to any of their education and work information, they click on Deny for 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. As new content is created (e.g., new pictures are taken), the user can set access rights (e.g., view) for this new content by associating them with existing same-as example friends. Or the user may establish a new policy by repeating the process outlined above.

3.3 Same-As Policy Management with Example Friend Selection

The visual policy uses three approaches for assisting users in selecting their same-as example friend: *Random*, *CNM Order*, and *Sample CNM Order*. Random presents friends to the user in random order. Both the CNM Order and Sample CNM Order approaches leverage the CNM network clustering algorithm. Our prototype clusters the user's social network graph creating CNM clusters of friends.

In CNM Order, we present the user's friends in CNM cluster order, i.e., all the friends in Cluster #1 are presented to the user followed by all the friends in Cluster#2, and so on. The first friend presented for each cluster is the friend with the highest degree (friend with the highest number of friend connections) in that cluster. This friend is the same-as example friend for that cluster. The premise is the highly connected friends are potentially more well known and thus easier to remember making them good candidates for same-as example friends. For example, Fig. 6 illustrates a user's social network graph that has three CNM clusters of friends. Friend A has the highest degree in Cluster #1 and, therefore, Friend A is presented to the user first as a recommendation for a same-as example friend. After Friend A is presented to the user, the remaining friends of Cluster #1 are presented for association with an already defined same-as example

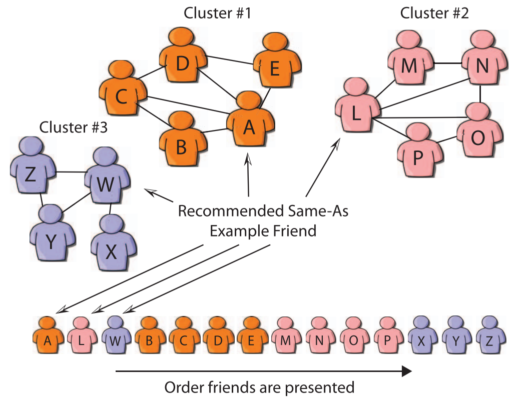


Fig. 7. Sample CNM Order.

friend or for assignment as a new same-as example friend. After all of Cluster #1 friends are presented, Cluster #2 friends are presented in a similar fashion, i.e., Friend L has the highest degree in Cluster #2 and thus is presented to the user as a possible candidate for a same-as example friend followed by the remainder of the friends in Cluster #2. This same process is repeated for all clusters.

The premise is by presenting the friends in CNM cluster order, the user can set the policy for the Same-As Example Friend and then associate all other similar friends with this Same-As Example Friend. The user's mental model is focused on one Same-As Example Friend at a time. After the policy is set for the Same-As Example Friend, the user can quickly ascertain that the stream of friends that follow may potentially be associated with this Same-As Example Friend.

In our second approach for assisting users in selecting their Same-As Example Friend, called Sample CNM Order, we present all of the friends with the highest degree within their cluster first. These friends are highly connected and are potentially more well known and, thus, easier to remember making them good candidates for Same-As Example Friends. Using the example social network graph depicted in Fig. 7, Sample CNM Order will present Friends A, L, and W first followed by the remainder of the friends from Cluster #1, followed by the reminder of the friends from Cluster #2, and then the remainder of the friends from Cluster #3. In Sample CNM Order, users enable their policies globally followed by policy assignment for each of their friends. The premise of this approach is that the user will set all their policies for all their Same-As Example Friends first and then quickly associate the stream of friends that follow with their respective Same-As Example Friend.

3.4 Prototype Architecture

We implemented two prototype Facebook applications: Group-based policy management (with Assisted Friend Grouping) and Same-As Policy Management (with Example Friend Selection). The applications are 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 applications, REST like Facebook APIs and Facebook Query Language are used to retrieve the user's profile and social connections. The collected data are transmitted over secure HTTPS-based APIs to our server and stored in a MySQL database. The applications build the participant's social graph, which is clustered using the CNM implementation provided by the Flare Toolkit Library. The application implements several additional functionalities, including user grouping, group policy specification, Same-As policy specification, and survey tools.

TABLE 1
User Study Tasks

Group Based Policy Management *(Facebook Application: Policy Manager)*	
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 *(Facebook Application: Policy Manager Same-As)*	
Task 4	Set permissions for friends using another friend's permissions as the model/example
Survey 2	Complete a brief survey for Task 4

4 User Study

In designing our user study [Approved IRB Protocol #11-08-01], we set out to answer the following research questions:

- Q1. Do proven clustering techniques align with user-defined relationship groups?
- Q2. Can proven clustering techniques assist users in grouping their friends more efficiently?
- Q3. What are users' perceptions of Assisted Friend Grouping techniques?
- Q4. Will a policy management approach based on leveraging a user's memory and perception of their friends outperform traditional group-based policy management approaches?
- Q5. Do different policy management approaches impact the conservativeness of a user's policy?
- Q6. 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?
- Q7. Can different friend selection techniques effectively aid users in picking example friends that are used in developing policy templates?

4.1 Design

To answer these research questions, we built four tasks and two surveys into our two prototype Facebook applications. 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 and Example Friend Selection. See Table 1.

In the first task (Task 1), the user is instructed to place 50 of their randomly selected friends into the 10 predefined 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

TABLE 2
Sampling of Survey Questions

Ease of Use	
Question 1	It was simple to use this system.
Question 2	Overall, I am satisfied with the ease of completing the tasks.
Question 3	It was easy to learn to use this system.
Readability	
Question 4	I easily understood who had access to what in my profile.
Question 5	The information was effective in helping me complete the tasks and scenarios.
Question 6	I could understand what my friends could access in my profile.
Flexibility	
Question 7	The system had enough flexibility in allowing me to set what my friends could access in my profile.
Question 8	This system has all the functions and capabilities I expect it to have.
Question 9	I could easily set what my friends could access in my profile.
Westin Privacy Sentiment	
Question 10	Users have lost all control over how personal information is collected and used by companies.
Question 11	Most businesses handle the personal information they collect about users in a proper and confidential way.
Question 12	Existing laws and organizational practices provide a reasonable level of protection for user privacy today.

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. Table 2 provides a sampling of the questions, which were presented to the user in a different order than they actually appear in the table. 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 that 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, for example, 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 second prototype Facebook application includes the fourth task and second survey. This 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. We divided the user participants into three groups, namely *Random*, *CNM Order*, and *Sample CNM Order*. For the Random population, the 50 friends were presented to the user in random order. For the CNM Order and Sample CNM Order populations, the 50 friends were presented to the user in CNM Order and Sample CNM Order, respectively, as described in Section 3.3. After the user selected their Same-As Example Friend, they then set appropriate profile object permissions for this example friend and assigned 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.

4.2 Participants

We recruited our user study participants from Amazon Mechanical Turk. 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 two prototype Facebook applications as two separate HITs. One HIT included the first three tasks and the first survey. The second HIT included the fourth task and second survey, as described in Section 4.1. To better control the quality of the recruited participants, we mandated that each worker have a minimum of 100 friends and a 95 percent HIT approval rating, or better. A HIT took approximately 10-15 minutes to complete, for which each worker was paid a fee of $1.00.

Our user study consisted of two populations: Group Based (145 participants) and Same-As (153 participants). The male/female ratio was approximately 6:4. Most of our user participants were young, fairly well educated, and active Facebook users. Approximately 54 percent (Group Based) and 60 percent (Same-As) were between the ages of 18 to 25. Almost 69 percent (Group Based) and 72 percent (Same-As) had between two and four years of college. Seventy-four percent of the Group-Based participants and 77 percent of Same-As participants used Facebook daily. In

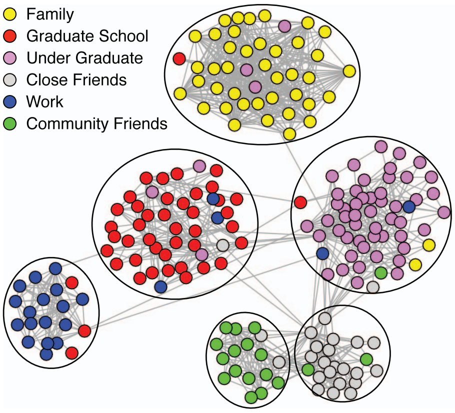


Fig. 8. Example cluster/group alignment.

TABLE 3
Not Assisted versus Assisted Friend Grouping

Measure	Not Assisted (μ, σ)	Assisted (μ, σ)	p-value
Adjusted Rand Index			
Unconcerned	(0.521, 0.05)	(0.573, 0.03)	0.218
Pragmatist	(0.625, 0.06)	(0.637, 0.06)	0.431
Fundamentalist	(0.677, 0.08)	(0.713, 0.08)	0.320
All	(0.627, 0.07)	(0.650, 0.07)	0.118
Grouping Time (seconds)			
Unconcerned	(175.2, 106.2)	(108.8, 29.0)	0.306
Pragmatist	(220.7, 45.9)	(178.9, 43.4)	< 0.001
Fundamentalist	(294.5, 45.2)	(202.0, 52.3)	< 0.001
All	(231.0, 60.1)	(180.2, 48.4)	< 0.001
User Perceptions (7 point Likert-scale)			
Ease of Use	(4.63, 0.85)	(5.33, 1.11)	< 0.001
Readability	(4.31, 0.72)	(5.32, 1.06)	< 0.001
Flexibility	(4.38, 0.66)	(5.33, 0.99)	< 0.001

addition, as part of the demographics portion of our survey, we collected Westin privacy sentiment information. Leveraging Westin's categorization method [24], users were labeled as Unconcerned Users, Pragmatists, and Fundamentalists based on their responses to Questions 10-12 (see Table 2). Unconcerned Users disagreed with Question 10 and agreed with Questions 11 and 12. Fundamentalists agreed with Question 10 and disagreed with Questions 11 and 12. All other users were categorized as Pragmatists. Our Group-Based population was made up of 6 percent Unconcerned Users, 73 percent Pragmatists, and 21 percent Fundamentalists. Our Same-As population was made up of 7 percent Unconcerned Users, 64 percent Pragmatists, and 29 percent Fundamentalists.

5 Study Results/Human Effects

The next sections detail our user study results and human effects for Assisted Friend Grouping, Same-As Policy Management, and Example Friend Selection.

5.1 Group-Based Policy Management with Assisted Friend Grouping

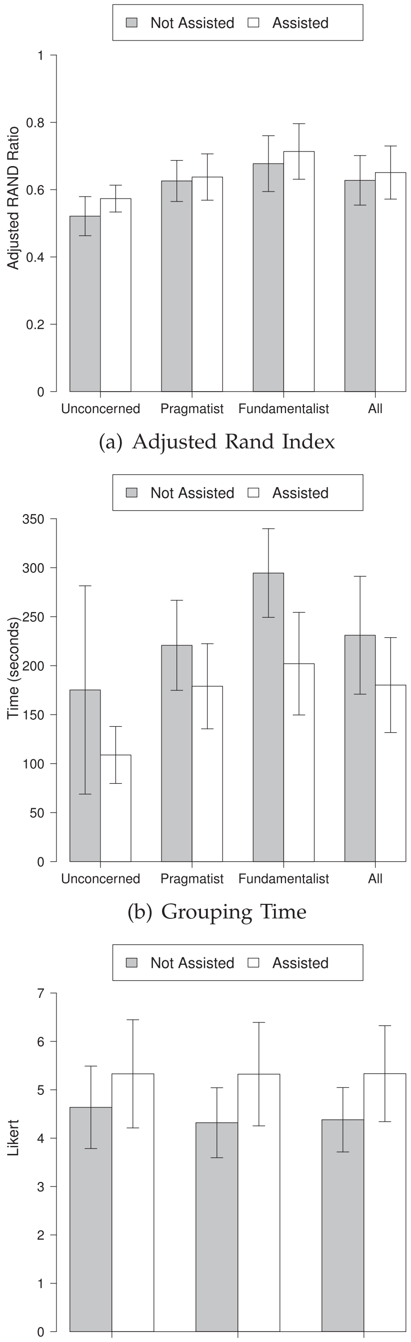
In evaluating our Assisted Friend Grouping Model, we set out to show that CNM will aid in grouping users' friends more efficiently for group-based policy management approaches. Our hypothesis is that CNM clusters roughly align with user-defined friend relationship groups. In the example illustrated in Fig. 8, 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, and so on. Fig. 8 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.

The Rand Index [33] is one of the standard metrics used to compare two partitions [28]. One shortcoming of the Rand Index is the expected value (what you would expect on average) of comparing two random partitions does not produce a constant value such as 0. The Adjusted Rand Index overcomes this shortcoming by ensuring the expected value of comparing two random partitions is constant [19]. We used the Adjusted Rand Index to measure the agreement between CNM clusters and user-defined relationship groups. The Adjusted Rand Index is the Rand Index adjusted to address partition agreement by chance. In other words, the Adjusted Rand Index is the difference of the Rand Index and its expected value. In general form, the Adjusted Rand Index can be described as $\frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex}$. 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.

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 6.3 relationship groups. Overall, our results showed an average Adjusted Rand Index of 0.627. This demonstrates that there is overlap and a level of alignment between CNM clusters and user-defined relationship groups. In looking just at Fundamentalist Users, we saw a higher level of alignment (Adjusted Rand Index $= 0.677$).

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 ($p = 0.118$). Refer to the Adjusted Rand Index section of Table 3 and Fig. 9a, 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.

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



(a) Adjusted Rand Index

(b) Grouping Time

(c) User Perceptions

Fig. 9. Not Assisted versus Assisted Friend Grouping.

time it took a 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 ($p = 0.306$). However, we did see statistical significance between the other categories of users: Pragmatists, Fundamentalists, and the population as a whole—all p-values were less than 0.001. Overall, using CNM, we saw a 22 percent reduction in time that it took a user to group 50 of their friends, 180.2 seconds (Assisted) versus 231 seconds (Not Assisted). Refer to the Grouping Time section of Table 3 and Fig. 9b. 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.

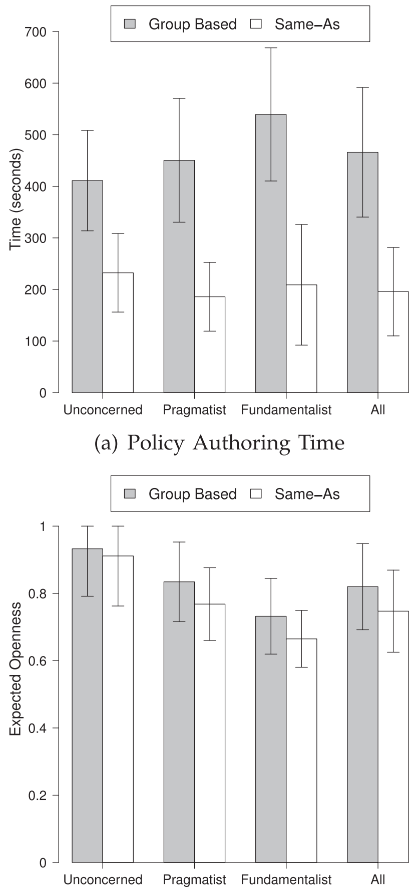
We also measured users' perceptions of the Not Assisted and Assisted Friend Grouping approaches, as described in Section 4.1. A t-test was used to compared the Not Assisted and Assisted populations. We found statistical significance in all user perception areas: Ease of Use, Readability, and Flexibility—all p-values were less than 0.001. Users found friend grouping easier to use when their friends were presented in CNM order. For Ease of Use, Not Assisted averaged 4.63 and Assisted averaged 5.33 on a 7 point Likert-scale. Readability and Flexibility also had similar results. Refer to the User Perceptions section of Table 3 and Fig. 9c. Overall, users had more positive perceptions of grouping their friends leveraging CNM than not having the assistance of CNM.

TABLE 4
Group Based versus Same-As

Measure	Group Based (μ, σ)	Same-As (μ, σ)	p-value
Policy Authoring Time (seconds)			
Unconcerned	(411.0, 97.3)	(232.3, 76.1)	0.001
Pragmatist	(450.3, 119.9)	(185.8, 66.6)	< 0.001
Fundamentalist	(539.4, 129.1)	(208.9, 116.8)	< 0.001
All	(465.9, 125.6)	(195.6, 85.6)	< 0.001
Policy Openness (see Definition 1)			
Unconcerned	(0.932, 0.141)	(0.911, 0.148)	0.769
Pragmatist	(0.834, 0.118)	(0.768, 0.108)	< 0.001
Fundamentalist	(0.732, 0.112)	(0.664, 0.084)	0.018
All	(0.819, 0.127)	(0.747, 0.122)	< 0.001

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 4 and illustrated in Fig. 10a. In analyzing these results, we found that there is statistical significance across all user categories, i.e., Unconcerned Users ($p = 0.001$), 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 twofold 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 to group them appropriately. Next, the user switches their attention to the group to set the group policy. Finally, the user switches their attention back to the friend to set any



(a) Policy Authoring Time

(b) Policy Openness

Fig. 10. Group Based versus Same-As Policy Management.

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.

We measured Policy Openness relative to a user's profile object (i.e., e-mail address) and found, for Unconcerned Users, no statistical significance between Group Based and Same-As ($p = 0.769$). 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 ($p < 0.001$), Fundamentalists ($p = 0.018$), and for the population as a whole ($p < 0.001$). Our findings are summarized in the Policy Openness section of Table 4 and Fig. 10b. 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.

TABLE 5
GP Based versus Same-As—User Perceptions

Measure	Group Based (μ, σ)	Same-As (μ, σ)	p-value
Ease of Use (7 point Likert-scale)			
Unconcerned	(5.20, 0.92)	(6.11, 0.59)	0.046
Pragmatist	(4.98, 1.03)	(6.06, 0.59)	< 0.001
Fundamentalist	(4.90, 1.15)	(5.94, 0.64)	< 0.001
All	(4.98, 1.04)	(6.03, 0.61)	< 0.001
Readability (7 point Likert-scale)			
Unconcerned	(4.97, 1.14)	(6.06, 0.60)	0.049
Pragmatist	(4.85, 0.94)	(6.04, 0.56)	< 0.001
Fundamentalist	(4.82, 1.02)	(5.89, 0.65)	< 0.001
All	(4.85, 0.96)	(6.00, 0.59)	< 0.001
Flexibility (7 point Likert-scale)			
Unconcerned	(5.05, 1.06)	(5.90, 0.72)	0.095
Pragmatist	(4.78, 1.02)	(6.00, 0.59)	< 0.001
Fundamentalist	(4.86, 1.12)	(5.75, 0.71)	0.002
All	(4.82, 1.03)	(5.92, 0.64)	< 0.001

Overall, users found Same-As easier to use than Group Based, 6.03 versus 4.98 on a 7-point Likert-scale, where 7 is Strongly Agree. We found statistical significance in our comparison ($p < 0.001$). Refer to the Ease of Use section of Table 5. Using Same-As over Group Based, we observed statistical significance and improved Ease of Use ratings for all user categories: Unconcerned Users, Pragmatists, and Fundamentalists. We attribute the improved ratings to reasons 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. It is interesting to note that Unconcerned Users averaged Ease of Use ratings higher than Pragmatists and Fundamentalists. Unconcerned Users do not 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.

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 5. 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 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 higher ratings to Same-As over Group Based, 5.92 versus 4.82. We found statistical significance for Pragmatists, Fundamentalists, and the population as a whole. However,

TABLE 6
Random versus CNM Order versus Sample CNM Order

Measure	Random (control) μ	CNM Order μ	Sample CNM μ	$F-Statistic$ $p-value$
Policy Authoring Time (seconds)				
	250.0	192.3	149.2	F(2,150)=21.65 $p < 0.001$
User Perceptions (7 point Likert-scale)				
Ease of Use	5.67	6.04	6.35	F(2,150)=18.68 $p < 0.001$
Readability	5.66	5.97	6.34	F(2,150)=19.67 $p < 0.001$
Flexibility	5.58	5.90	6.26	F(2,150)=16.52 $p < 0.001$

we did not find significance between the two approaches for Unconcerned Users. Refer to the Flexibility section of Table 5. In access control terms, both Group Based and Same-As have similar 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 and was perceived to be more flexible.

5.3 Same-As Policy Management with Example Friend Selection

We also evaluated the three approaches used by the Same-As Policy Management visual policy editor for assisting users in selecting their Same-As Example Friend: *Random*, *CNM Order*, and *Sample CNM Order*, as described in Section 3.3. Using analysis of variance (ANOVA), we measured the effects of the three approaches. Our results are summarized in Table 6.

In evaluating authoring time, we observed a 23 percent reduction in the time it took a user to author a policy leveraging CNM Order (192.3 seconds) versus Random (250 seconds). Fig. 11 displays the policy authoring time results in the form of box plots, where the top and bottom of the boxes are the first and third quartiles, respectively, and the band near the middle of the box is the median. We see statistical significance among the three groups ($p < 0.001$) with the F-Statistic (21.65) greater than 3.06 for a probability of 95 percent. We also ran a pairwise comparison leveraging the Bonferroni correction, where we observed statistical significance across all pairings.

CNM Order allows users to author policies faster because we recommend highly connected friends as Same-As Example Friends. The most highly connected friend of a cluster is presented first and is more likely to be selected as a Same-As Example Friend. This highly connected friend is potentially more well known and, thus, easier to remember making them good candidates for Same-As Example Friends. After the policy is set, the stream of friends presented next is of the same cluster and potentially the same relationship group and policy template. The user's mental model is focused on one Same-As Example Friend where they can quickly associate, if appropriate, the stream of friends that follow with this Same-As Example Friend. This process repeats itself for each of the user's clusters.

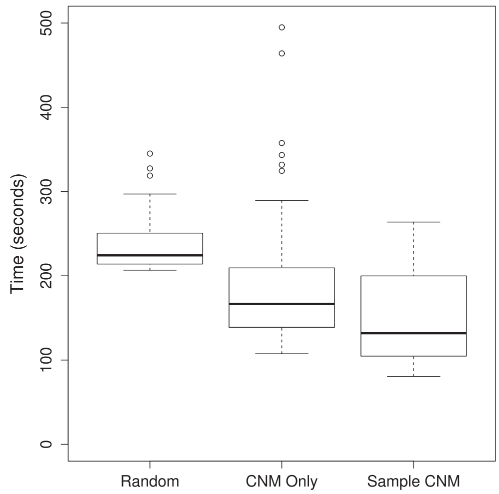


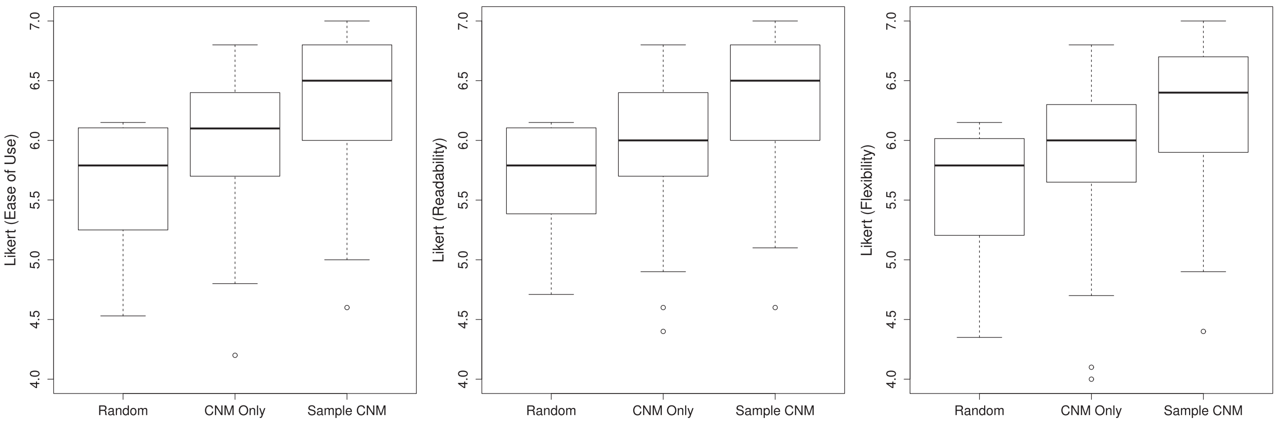
Fig. 11. Policy authoring time.

Sample CNM Order (149.2 seconds) outperformed CNM Order with a 22 percent reduction in policy authoring time. In addition, Sample CNM Order outperformed Random with a 40 percent reduction in policy authoring time. With Sample CNM Order, all the user's clusters' most highly connected friends are presented first for policy authoring and then the remaining members of each cluster are presented in cluster order for association with the appropriate Same-As Example Friend. With this Example Friend Selection technique, the user sets all their policy templates (Same-As Example Friends) first and then associates appropriate friends with each policy template. Users were able to author policies much faster leveraging this technique over Random and CNM Order.

In measuring user perceptions of the three approaches for selecting the Same-As Example Friend, we observed that Sample CNM Order was more positively perceived than Random and CNM Order. Sample CNM Order was found to be easier to use (6.35 on a 7-point Likert-scale), more readable (6.34) and more flexible (6.26). See Fig. 12. We found statistical significance ($p < 0.001$) across the three areas measured (Ease of Use, Readability, Flexibility) with all F-Statistics greater than 3.06 for a probably of 95 percent. We also ran a pairwise comparison leveraging the Bonferroni correction where we observed statistical significance across all pairings. Sample CNM Order, where the user authors all the policies for their Same-As Example Friends first, outperformed both Random and CNM Order for both policy authoring time and user perceptions.

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 burden associated with populating relationship groups for large friends sets. Assisted Friend Grouping, which demonstrated measurable agreement between CNM clusters and user-defined relationship groups, alleviates this



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

Fig. 12. Random versus CNM Order versus Sample CNM Order—User Perceptions.

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-based policy management. It provides a similar 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 xsreadable policies that are not complex and difficult to understand. In addition, users can compose these policies in less than 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, users are authoring 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.

In evaluating different friend selection techniques for aiding users in picking example friends (Same-As Example Friends), we found that both techniques introduced (CNM Order and Sample CNM Order) outperformed the Random approach. Each new technique reduced the time it took to author a policy. In addition, users' perceptions were higher over the Random approach. Presenting friends in CNM cluster order using either technique (CNM Order or Sample CNM Order) potentially cuts down on the amount of "searching" a user must do to find the "right" Same-As Example Friend. We present the friends in an order that is potentially meaningful to the user. As such, we would expect to have faster policy authoring times and improved user perceptions.

In evaluating Sample CNM Order versus CNM Order, we see the former outperforming the latter in both policy authoring time and user perceptions. Sample CNM Order allows a user to build their global policy set upfront. After which, they can quickly assign appropriate friends to each policy. In leveraging this approach, users were able to author policies more quickly than the other two approaches. In addition, users had higher perceptions of Sample CNM Order.

Finally, our results validate that Unconcerned Users author more open policies and are less concerned about policy authoring flexibility than pragmatists and fundamentalists. Or, conversely, we found Fundamentalists authored the most conservative policies and desired the most authoring flexibility. As would be expected, Pragmatists fell in the middle. Leveraging these findings, privacy management models can be enhanced by detecting and using user privacy sentiment. For example, Unconcerned Users who author more open policies may leverage a less flexible coarse-grained privacy management approach. Whereas a Fundamentalist, who authors more conservative policies, will find a fine-grained approach better suited for meeting their privacy needs. Privacy management models can be further refined and enhanced by detecting and leveraging user privacy sentiment in managing access to user privacy information.

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 to be placed in the "best" group. Clearly, there are examples where we would expect a friend to be in multiple groups, for example, 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 future research. Also, our user study participants were comprised of *Workers* recruited 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.

7 Related Work

Yuksel et al. [37] 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.

Fang and LeFevre [14] outline an approach, using machine learning, to describe a user's privacy preferences. In essence, they build a training set by asking the user to label (allow or deny) a subset of friends relative to a specific object. In addition, the training set contains other friend specific attributes—primarily age, gender, and social network community. We also leverage community (CNM clusters) in our enhanced privacy models. However, our primary contribution relative to community detection is demonstrating, through empirical data, measurable agreement between user-defined relationship groups and social network communities.

Carminati et al. [6], [7] propose an access control framework and language for social networks that describes user profiles, relationships among friends and profile objects. Gates [18] introduces the term relationship-based access control (ReBAC). The premise of ReBAC is that access control decisions are based on the relationship of the object's owner and the subject, versus, for example, the role of the subject as in RBAC. Fong [16] formalizes the ReBAC model and introduces an approach to capture the context of the relationship. In addition, he introduces a policy language. Fong and Siahaan [17] propose extensions to the relational policy language. Cheng et al. [8] propose a user-to-user relationship-based access control model for social networks with a supporting policy specification language.

Mazzia et al. [29] introduce a policy visualization tool that displays privacy settings for user specific subgroups of friends within social networks. Besmer et al. [4] analyze the impacts of community information on access control policy decisions within social networks. Lipford et al. [27] 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 [11], [12] and the ill effects of work/task interruption [20], [10]. Same-As Policy Management leverages concentrated memory recognition of friends using a visual policy editor to manage privacy in online social networks.

8 Conclusion and Future Work

In this paper, we enhance existing and introduce new privacy management models, in addition to measuring their human effects. First, we present an enhancement to traditional group-based policy management, which assists users in grouping their friends more efficiently. With Assisted Friend Grouping, we found measurable agreement between clusters and user-defined relationship groups. Second, we introduce Same-As Policy Management, which leverages users' memory and opinion of their example friends to set policies for other similar friends. Finally, we introduce two techniques for aiding users in selecting their example friends. By associating policy templates with friends versus group labels, Same-As Policy Management allowed users to author policies more efficiently and was more positively perceived over traditional group-based policy management. In addition, by leveraging our user study results, privacy management models can be further enhanced by detecting and leveraging user privacy sentiment. Based on a user's privacy sentiment, the privacy management model can be tailored. For example, for unconcerned users, a more coarse-grained privacy management model could be leveraged and for Fundamentalists, a more fine-grained approach could be used.

Our future work plans include running additional studies and comparing the two CNM-based policy management model enhancements (Assisted Friend Grouping and Example Friend Selection) in terms of policy definition, openness, and their human effects. In addition, we plan to further investigate patterns in alignment of clusters and user-defined relationship groups. We also plan to develop a prototype that leverages user privacy sentiment for the mass customization of a privacy management model.

Acknowledgments

This research was partially supported by grants from the US National Science Foundation (NSF-CNS-0831360, NSF-CNS-1117411) and a Google Research Award. The authors would also like to thank Adharsh Desikan for his contributions to this research.

References

[1] R. McMillan, "Google Buzz Criticized for Disclosing Gmail Contacts," http://www.pcworld.com/article/189081/google_buzz_criticized_for_disclosing_gmail_contacts.html, 2010.

[2] A. Acquisti and R. Gross, "Imagined Communities: Awareness, Information Sharing and Privacy on the Facebook," *Proc. Sixth Int'l Conf. Privacy Enhancing Technologies (PET '06),* 2006.

[3] A. Acquisti and J. Grossklags, "Privacy and Rationality in Individual Decision Making," *IEEE Security & Privacy,* vol. 3, no. 1, pp. 26-33, Jan./Feb. 2005.

[4] A. Besmer, J. Watson, and H.R. Lipford, "The Impact of Social Navigation on Privacy Policy Configuration," *Proc. Symp. Usable Privacy and Security,* 2010.

[5] J. Bonneau and S. Preibusch, "The Privacy Jungle: On the Market for Data Protection in Social Networks," *Proc. Workshop the Economics of Information Security (WEIS '09),* 2009.

[6] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "A Semantic Web Based Framework for Social Network Access Control," *Proc. Symp. Access Control Models and Technologies,* 2009.

[7] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B.M. Thuraisingham, "Semantic Web-Based Social Network Access Control," *Computers & Security,* vol. 30, pp. 108-115, 2011.

[8] Y. Cheng, J. Park, and R.S. Sandhu, "A User-to-User Relationship-Based Access Control Model for Online Social Networks," *Proc. 26th Ann. IFIP WG 11.3 Conf. Data and Applications Security and Privacy,* 2012.

[9] A. Clauset, M. Newman, and C. Moore, "Finding Community Structure in Very Large Networks," *Physical Rev. E,* vol. 70, p. 066111, 2004.

[10] E. Cutrell, M. Czerwinski, and E. Horvitz, "Notification, Disruption, and Memory: Effects of Messaging Interruptions on Memory and Performance," *Proc. Conf. Human Computer Interaction,* 2001.

[11] R. Dhamija and A. Perrig, "Deja Vu: A User Study Using Images for Authentication," *Proc. USENIX Security Symp.,* 2000.

[12] P. Dunphy, A.P. Heiner, and N. Asokan, "A Closer Look at Recognition-Based Graphical Passwords on Mobile Devices," *Proc. Symp. Usable Privacy and Security,* 2010.

[13] C. Dwyer, S.R. Hiltz, and K. Passerini, "Trust and Privacy Concern within Social Networking Sites: A Comparison of Facebook and MySpace," *Proc. Am. Conf. Information Systems (AMCIS '07),* 2007.

[14] L. Fang and K. LeFevre, "Privacy Wizards for Social Networking Sites," *Proc. Conf. World Wide Web,* 2010.
[15] D. Ferraiolo and R. Kuhn, "Role-Based Access Control," *Proc. Nat'l Computer Security Conf.,* 1992.
[16] P.W. Fong, "Relationship-Based Access Control: Protection Model and Policy Language," *Proc. Conf. Data and Application Security and Privacy,* 2011.
[17] P.W.L. Fong and I. Siahaan., "Relationship-Based Access Control Policies and Their Policy Languages," *Proc. Symp. Access Control Models and Technologies,* 2011.
[18] C.E. Gates, "Access Control Requirements for Web 2.0 Security and Privacy," *Proc. Workshop Web 2.0 Security and Privacy,* 2007.
[19] L. Hubert and P. Arabie, "Comparing Partitions," *J. Classification,* vol. 2, pp. 193-218, 1985.
[20] S.T. Iqbal and B.P. Bailey, "Investigating the Effectiveness of Mental Workload As a Predictor of Opportune Moments for Interruption," *Proc. Computer Human Interaction Extended Abstracts on Human Factors in Computing Systems (CHI '05),* 2005.
[21] 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," *Proc. Conf. Computer Supported Cooperative Work (CSCW '04),* 2004.
[22] S. Jones and E. O'Neill, "Feasibility of Structural Network Clustering for Group-Based Privacy Control in Social Networks," *Proc. Symp. Usable Privacy and Security,* 2010.
[23] H. Krasnova, O. Günther, S. Spiekermann, and K. Koroleva, "Privacy Concerns and Identity in Online Social Networks," *Identity in the Information Soc.,* vol. 2, no. 1, pp. 39-63, 2009.
[24] P. Kumaraguru and L.F. Cranor, "Privacy Indexes: A Survey of Westin's Studies," Technical Report CMU-ISRI-5-138, Inst. for Software Research Int'l, School of Computer Science, Carnegie Mellon Univ., 2005.
[25] 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,* vol. 8, pp. 440-454, 2004.
[26] K. Lewis, J. Kaufman, and N. Christakis, "The Taste for Privacy: An Analysis of College Student Privacy Settings in an Online Social Network," *J. Computer-Mediated Comm.,* vol. 14, pp. 79-100, 2008.
[27] H.R. Lipford, J. Watson, M. Whitney, K. Froiland, and R.W. Reeder, "Visual versus Compact: A Comparison of Privacy Policy Interfaces," *Proc. SIGCHI Conf. Human Factors in Computing Systems,* 2010.
[28] C.D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval.* Cambridge Univ. Press, 2008.
[29] A. Mazzia, K. LeFevre, and E. Adar, "The PViz Comprehension Tool for Social Network Privacy Settings," Technical Report CSE-TR-570-11, Univ. of Michigan, 2011.
[30] M.E.J. Newman, "Fast Algorithm for Detecting Community Structure in Networks," *Physical Rev. E,* vol. 69, article 066133, 2004.
[31] P.A. Norberg, D.R. Horne, and D.A. Horne, "The Privacy Paradox: Personal Information Disclosure Intentions versus Behaviors," *J. Consumer Affairs,* vol. 41, pp. 100-126, 2007.
[32] J.S. Olson, J. Grudin, and E. Horvitz, "A Study of Preferences for Sharing and Privacy," *Proc. Computer Human Interaction (CHI '05) Extended Abstracts on Human Fcators in Computing Systems,* 2005.
[33] W. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *J. Am. Statistical Assoc.,* vol. 66, pp. 846-850, 1971.
[34] R. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, "Role-Based Access Control Models," *Computer,* vol. 29, no. 2, pp. 38-47, Feb. 1996.
[35] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The Nist Model for Role-Based Access Control: Towards a Unified Standard," *Proc. Workshop Role-Based Access Control,* 2000.
[36] K. Strater and H.R. Lipford, "Strategies and Struggles with Privacy in an Online Social Networking Community," *Proc. British HCI Group Ann. Conf. People and Computers: Culture, Creativity, Interaction,* 2008.
[37] A.S. Yuksel, M.E. Yuksel, and A.H. Zaim, "An Approach for Protecting Privacy on Social Networks," *Proc. Conf. Systems and Networks Comm.,* 2010.

Gorrell P. Cheek received the master's degree in computer science from the University of Virginia and the MBA degree from Wake Forest University. He is currently working toward the PhD degree in the Department of Software and Information Systems, College of Computing and Informatics, University of North Carolina at Charlotte. His research interests include information security and privacy, with a focus area of access-control methodologies in online social networks. He is a member of the IEEE, the IEEE Computer Society, and the ACM.

Mohamed Shehab received the PhD degree in computer engineering from Purdue University. He is an associate professor in the Department of Software and Information Systems, College of Computing and Informatics, University of North Carolina at Charlotte. He is the director of the Laboratory of Information Integration Security and Privacy. His research interests lie in network and information security, mobile applications, web services, and peer-to-peer environments. He is a member of the IEEE, the IEEE Computer Society, and the ACM.