2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Usable Object Management Approaches for Online Social Networks

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

***Abstract*—Online social networks have seen tremendous adoption and growth in recent years. Most attention, in access control literature, has been placed on abstracting and managing the large numbers of subjects or friends within these online social networks. Usable approaches for managing large amounts of objects, in the form of privacy information and content, have lagged. We introduce two approaches for object management. We extend our previous work to accommodate for object grouping and we introduce Same-As Object Management, which provides for a more usable object management approach that is effective, efficient and satisfying to the user. Same-As Object Management leverages a user's memory and perception of their objects for setting permissions for other similar objects. We implemented our model in an online social network and conducted a user study whose results are encouraging.**

I. Introduction

Research has found that managing access to online information (both privacy and content) is traditionally manual, complex, difficult and time consuming [3], [7], [10]. Additional research points to the long sought after goal and importance of usable security [1], [9], [16]. But, *usability* and *security* often have competing objectives. According to ISO 9241-11 (1998), *usability* is the "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." ISO 17799 (2005) states that "*security* is achieved by implementing a suitable set of controls... to ensure that the specific security and business objectives... are met." How do we build suitable access control frameworks that are effective, efficient and satisfying to the end user? These frameworks must adhere to all three criteria. If they are suitable to the end user but don't provide effective controls, then security is not achieved. Conversely, access control frameworks not only have to be effective and efficient, they must also be satisfying. If they are not, they are unlikely to be used and therefore become ineffective. Saltzer and Schroeder [13] emphasize the importance of *psychological acceptability* as a key protection mechanism design principle. "It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly. Also, to the extent that the user's mental image of his protection goals matches the mechanisms he must use, mistakes will be minimized."

Access control frameworks must be easy to use. Users must be able to manage access to their online information in a simple and intuitive way that aligns with their intentions. In addition, these frameworks must be designed such that they adhere to a user's mental image or model for managing and controlling access to their online information. Jones et al. [5] describe mental models as "personal, internal representations of external reality that people use to interact with the world around them; [they] are used to reason and make decisions and can be the basis of individual behaviors." Specific to access control frameworks, a user's mental model is their understanding of how access to their online information is managed and controlled. It is not necessarily derived from formal instruction or training of the framework. Even so, the framework's capabilities should align, as much as possible, to the user's mental model. For example, a user's mental model of an access control policy should align with how the online social network evaluates that policy. If the user's intent is to limit access to a set of sensitive pictures to just family members, the online social network should evaluate and enforce that policy accordingly. The more alignment of an access control framework with a user's mental model, the less likely for policy errors and unintended information leakage.

More usable object management approaches for online social networks are needed. Traditionally, usability improvements are made in two areas: 1) changes to the underlying access control model and 2) enhancements to the user interface [12]. Our contribution is three-fold:

- For the purposes of access control, we propose a new object management approach, called Same-As Object Management. Our approach is effective and efficient in that it allows users to organize their objects in a straightforward manner. Same-As Object Management demonstrated improved expressiveness and performance, in addition to more conservative policies, over more traditional grouping based approaches.
- We propose a policy management user interface for Same-As Object Management that is satisfying to the user. Our visual policy editor is easy to use and aligns with the user's mental model for managing access to their online privacy information and content. It demonstrated improved user perceptions over traditional grouping based policy management interfaces.
- We implemented a prototype Facebook application and conducted a user study evaluating our improvements to object management approaches in online social networks.

II. Background

Access control systems regulate the actions that subjects can take on objects. Subjects (e.g., friends) are the actors that invoke an action or access mode (e.g., read) on a user's object (e.g., privacy information like date of birth or content like a

ASONAM'13, August 25-29, 2013, Niagara, Ontario, CAN

picture). An authorization is a tuple $< s, o, a >$, where s is a subject, o is an object and a is an access mode. Subjects and objects are dynamic. Friends come and go. User content is added, updated and deleted on a regular basis. Users must maintain up to date and accurate authorization lists. This can be a daunting task. If there are m subjects, n objects and p access modes, then there are potentially $m \times n \times p$ authorizations. For example, if a user has 130 friends, 90 pieces of content and one access mode (read), the number of potential authorizations he must maintain is 11,700. Security administration, including granting / revoking authorizations (permissions), is very challenging.

One approach that has been taken to alleviate the burden of managing large numbers of authorizations is the implementation of role based access control (RBAC) [4], [14], [15]. Role based access control creates a level of abstraction for subjects by introducing a role which is a container that has functional meaning, e.g., a specific job within an enterprise. Object permissions are assigned to roles. Roles are then populated with subjects who are granted the object permissions associated with the role(s) in which they belong. This level of abstraction alleviates the burden of managing large numbers of subject to object permissions assignments.

In our previous research, we introduced Same-As Subject Management, which leverages a user's memory and opinion of their friends to set policies for other similar friends [2]. A visual policy editor uses friend recognition and minimal task interruption to obtain reductions in policy authoring times. In addition, Same-As Subject Management was positively perceived by users over traditional group based policy management approaches.

RBAC and Same-As Subject Management focus on abstracting the complexity of managing large numbers of subjects for the purposes of security administration. Most attention, in access control literature, has been placed on the subject side of the authorization equation. More limited research has focused on the object side of the authorization equation, i.e., how to manage large sets of objects for the purposes of controlling access. The basic premise of access control systems is to protect and control access to sensitive resources / assets (i.e., objects). Just as subjects are dynamic, so are objects. Objects are created. Their properties change, e.g., sensitivity level, size, etc. They ultimately are retired or deleted. Managing the full life-cycle of large numbers of objects can be complex and difficult.

Moyer and Abamad extend RBAC by introducing Generalized Role Based Access Control [11]. GRBAC proposes environment and object roles. Environment roles describe environmental conditions on which access control decisions can be made, e.g., temporal, system load, etc. Object roles are similar to subject roles in that they provide a level of abstraction for objects. Objects can be grouped based on some common property, e.g., sensitivity level, creation date, size, etc. Thus, by grouping subjects and objects, the number of authorizations can be greatly reduced. If there are j subject groups ($j \leq m$), k object groups ($k \leq n$) and p access modes, then there are potentially $j \times k \times p \leq m \times n \times p$ authorizations. See Figure 1. For example, if a user has 10 friend groups, 20 object groups and one access mode (read), the number of authorizations he must maintain is 200. Security administration starts to become more manageable with subject / object grouping access control models.

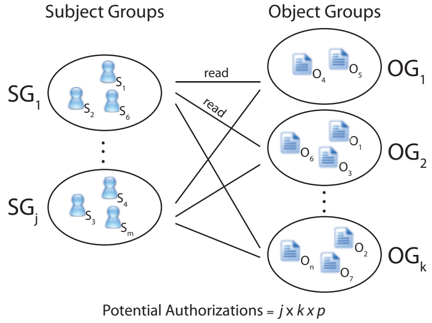


Fig. 1. Subject Grouping / Object Grouping

III. Object Management Approaches

For the purposes of access control in online social networks, we propose two new approaches for managing objects that are effective, efficient and, with the aid of a visual policy editor, satisfying to the user. We extend our previous research, called Same-As Subject Management [2], to allow for object grouping. In addition, we introduce a new approach for managing objects, called Same-As Object Management, which allows users to manage their objects and set access control policies in a straightforward manner.

A. Same-As Subject Management with Object Grouping

Most online social networks provide some means for grouping subjects, e.g., Facebook *Friend Lists* and Google+ *Circles*. However, most do not provide a means for grouping objects. Same-As Subject Management only allows for permissions to be set for individual objects. We extend Same-As Subject Management to accommodate for the setting of permissions for object groups. Objects can be grouped by their specific properties, e.g., size, creation date, type, event, location, sensitivity level, etc. Once the objects are grouped, the traditional Same-As Subject Management steps are followed. The user is asked to select a representative subject (Same-As Example Subject), set object *group* permissions and assign other similar subjects the same set of object *group* permissions. This process is repeated for each of the user's representative subjects. Figure 2 illustrates our model.

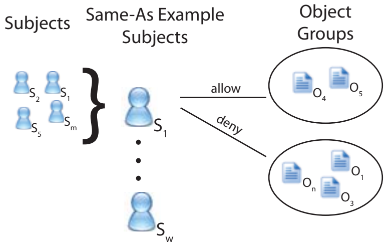


Fig. 2. Same-As Subject Management with Object Grouping

With our prototype Facebook application, the user is asked to group approximately 15 of their randomly selected pictures (objects) into four predefined and up to four more user defined object sensitivity groups. The predefined groups are labeled *Public* (Viewable by everyone), *Private* (Viewable by most, but not all), *Sensitive* (Viewable by select friends) and *Highly*

Sensitive (Viewable by a very select small set of friends). These labels were designed in part from the classification schemes used by many governmental organizations (e.g., *Unclassified*, *Confidential*, *Secret* and *Top Secret*) and those used in the commercial sector, e.g., as described in RFC 3114. Once the objects are grouped, the traditional Same-As Subject Management steps are followed. Using the Same-As Subject Management policy editor, the user selects a friend (Same-As Example Subject) that representatives a subset of all their friends. The user then sets permissions for this representative friend by associating object groups they are allowed to access. After the permissions are set, other like friends are assigned to the same policy.

B. Same-As Object Management

Same-As Object Management leverages the same basic principles as Same-As Subject Management but policies are built around representative objects instead of subjects. With Same-As Object Management, the user first groups their subjects. After which, the user is asked to select a representative object (Same-As Example Object), set subject group permissions and assign other similar objects the same set of subject group permissions. This process is repeated for each of the user's representative objects. Figure 3 illustrates our model.

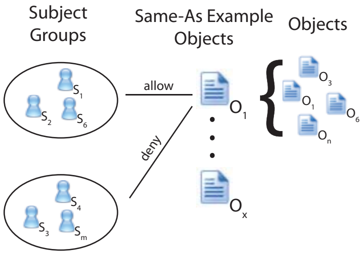


Fig. 3. Same-As Object Management

With our prototype Facebook application, the user is asked to group approximately 30 of their randomly selected friends (subjects) into ten predefined friend groups: *Family, Close Friends, Graduate School, Under Graduate School, High School, Work, Acquaintances, Friends of Friend, Community, and Other*. These groups were selected based on the work of Jones et al. [6]. They propose that users group their friends, for privacy management purposes, based on six criteria: Social Circles, Tie Strength, Temporal Episodes, Geographical Locations, Functional Roles and Organizational Boundaries. Our friend groups were selected to reflect these criteria.

Using the Same-As Object Management policy editor, the user selects a picture (Same-As Example Object) that representatives a subset of all their pictures. See larger blue circle labeled "W" in Figure 4. This picture represents some subjective meaning of sensitivity to the user, of which other pictures of a similar sensitivity level can be associated. For example, Bob may select a picture that depicts him drinking alcohol as the representative picture that he uses to associate other similar pictures of him drinking alcohol. Bob considers pictures of him drinking alcohol to be of a certain sensitivity level and wants to limit who can view these pictures. The picture should be easy to remember and is the representative for other like or similar pictures.

Next, the user assigns the appropriate friend group permissions for this representative picture. The user can allow / deny access to any friend group by clicking on the group to toggle between allow / deny permissions. If the user doesn't want a specific friend group to have access to the representative picture, they merely click on that friend group and the group will be grayed out. This indicates that access is not allowed. For example, Bob may allow his *Family* group to view pictures of him drinking alcohol, but deny viewing rights for his *Work* group. The default permissions are set to deny access.

After the permissions are set, other like pictures are assigned to the policy. The visual policy editor presents to the user their picture set, where the user can associate a picture to an already defined representative picture. Or, the user can select a picture as a new representative picture, thereby setting a new policy, which other similar pictures would be assigned. This process repeats itself for the user's entire picture set. For example in Figure 4, our visual policy editor depicts Same-As Example Object "Z" as having access to the *Family* and *High School* friend groups. (The remaining friend groups are grayed out and, therefore, access is denied). All the similar pictures circled in red inherit these same permissions.

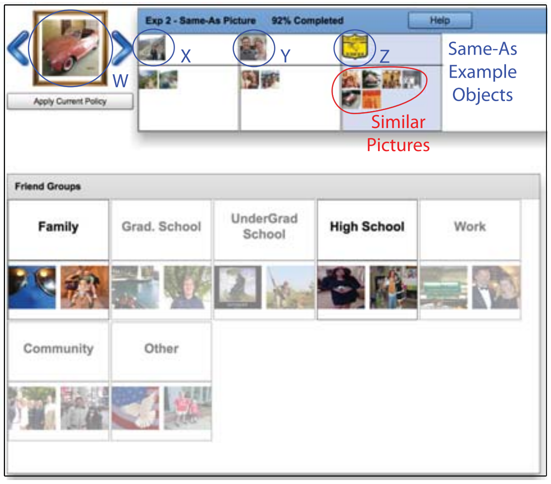


Fig. 4. Same-As Object Management Visual Policy Editor User Interface

C. Prototype Architecture

We implemented Facebook application prototypes of Same-As Subject Management with Object Grouping and Same-As Object Management. The Facebook application is hosted on our server. The backend 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 implements several functionalities including friend grouping, picture grouping, group policy specification, Same-As policy specification and survey tools.

IV. User Study

In designing our user study,[1] we set out to answer the following research questions:

[1] Approved Institution Review Board Protocol #11-08-01

Q1. Do our object management approaches allow for more expressive policies?

Q2. Do our object management approaches outperform more traditional approaches?

Q3. Will using our object management approaches lead to more conservative policies?

Q4. What are user's perceptions of our object management approaches?

A. Design

In order to answer these research questions, we designed a within subjects user study consisting of three experiments. To avoid ordering bias, the experiments were presented in random order to the study participants. These three experiments were implemented as part of our prototype Facebook application. The first experiment was designed to evaluate traditional subject / object grouping access control models (refer back to Figure 1). This experiment has three tasks, as indicated in Table I. First, the user was instructed to group approximately 30 of their randomly selected friends into ten predefined friend groups. After which, the user was asked to group approximately 15 of their randomly selected pictures in up to four predefined and four user defined (optional) sensitivity groups. Finally, the user was asked to select allow / deny access permissions specifying which subject groups have access rights to each object group. We measured how many unique policy templates the user created, how long the user took to author all their policies (to include grouping activities) and the conservativeness of their policy set. Upon completion of the three tasks for Experiment 1, the user completed a brief survey designed to capture their perceptions of subject / object grouping access control models.

The second experiment was designed to evaluate Same-As Subject Management with Object Grouping, as described in Section III-A. This experiment has two primary tasks. See Table I. In the first task (Task 4), the user was instructed to group approximately 15 of their randomly selected pictures in up to four predefined and four user defined (optional) sensitivity groups. In the second task (Task 5), the user was instructed, for a subset of their friends (approximately 30 randomly chosen ones), to select a Same-As Example Subject, set appropriate allow / deny access 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 how many unique policy templates the user created, how long the user took to author all their policies (to include grouping activities) and the conservativeness of their policy set. After completing Tasks 4 and 5, the user completed a second survey identical to the survey presented in Experiment 1.

The third experiment was designed to evaluate Same-As Object Management, as described in Section III-B. This experiment also has two primary tasks. See Table I. In the first task (Task 6), the user was instructed to group approximately 30 of their randomly selected friends in up to ten predefined friend groups. In the second task (Task 7), the user was instructed, for a subset of their pictures (approximately 15 randomly chosen ones), to select a Same-As Example Object, set appropriate allow / deny access permissions for this example picture and assign the policy to appropriate like or similar pictures. This step was repeated as necessary, i.e., for as many unique policies the user would like to assign for their picture set. We measured how many unique policy templates the user created, how long the user took to author all their policies (to include grouping activities) and the conservativeness of their policy set. After completing Tasks 6 and 7, the user completed a third survey identical to the survey presented in Experiment 1.

TABLE I. USER STUDY EXPERIMENTS

Experiment 1 – Subject / Object Grouping	
Task 1	Group subjects
Task 2	Group objects
Task 3	Set permissions
Survey 1	Complete a brief survey for Tasks 1-3
Experiment 2 – Same-As Subject Management w/ Object Grouping	
Task 4	Group objects
Task 5	Set permissions for subjects using another subject's permissions as the model / example
Survey 2	Complete a brief survey for Tasks 4-5
Experiment 3 – Same-As Object Management	
Task 6	Group subjects
Task 7	Set permissions for objects using another object's permissions as the model / example
Survey 3	Complete a brief survey for Tasks 6-7

B. Participants

We recruited our user study participants from Amazon Mechanical Turk. Amazon Mechanical Turk is a crowd sourcing marketplace. *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 three experiments and surveys as described in Section IV-A. We also mandated that each worker have a minimum of 100 friends, 15 pictures and a 95% HIT approval rating or better. A HIT took approximately 10 minutes to complete, for which each worker was paid a fee of $1.00. A total of 99 participants successfully completed the user study.

Of the 99 participants, 61 were male and 38 were female. Most of our user participants were young, fairly well educated and active Facebook users. 70% were between the ages of 18 to 25. 80% had between two and four years of college. Almost 87% used Facebook daily. In addition, as part of the demographics portion of our survey, we collected Westin privacy sentiment information [8], which allowed us to segment the user participants into three categories: Unconcerned users (9%), Pragmatist (64%) and Fundamentalist (27%).

V. STUDY RESULTS

We evaluated the three access control models: Subject / Object Grouping – hereafter referred to as Grouping Based (Experiment 1), Same-As Subject Management with Object Grouping – hereafter referred to as Same-As Subject (Experiment 2) and Same-As Object Management – hereafter referred to as Same-As Object (Experiment 3). Using analysis of variance (ANOVA), we measured the effects of the three approaches. For the Unconcerned user population, we observed no statistical significance for the three experiments across all measurements. The remaining results are summarized in the tables and figures that follow.

a) Number of Policy Templates: We measured how many unique policy templates a user created as part of each experiment. A policy template is a collection of authorizations created by the user with semantic meaning established by the user. For Grouping Based, unique policy templates equates to the number of sensitivity groups the user leverages. For Same-As Subject, unique policy templates equates to the number of Same-As Example Subjects the user creates and similarly for Same-As Object, where the number of unique policy templates equates to number of Same-As Example Objects the user creates.

For Fundamentalists, there is no statistical significance $(p = 0.06)$ with regard to the number of policy templates the user generates. See Number of Policy Templates section of Table II. We do see statistical significance for Pragmatists $(p < 0.01)$ and the population as a whole $(p < 0.01)$. The *F-Statistics* are greater than 3.04 (Pragmatist) and 3.02 (All) for a probability of 95%. We also ran a pairwise comparison leveraging the Bonferroni correction and observed no statistical significance between Same-As Subject and Same-As Object. However, we do see statistical significance between Grouping Based and Same-As Subject and Grouping Based and Same-As Object. Table III summarizes the results of the pairwise comparison with green indicating significance $(p < 0.05)$ and red otherwise. Same-As Subject and Same-As Object create approximately four policy templates versus approximately three for Grouping Based. Figure 5(a) displays the number of policy templates by experiment in the form of a box plot, where the top and bottom of the box is the first and third quartiles respectively and the band near the middle of the box is the median.

TABLE II. GROUPING BASED VS. SAME-AS SUBJECT VS. SAME-AS OBJECT

Measure	**GB** μ	**SaS** μ	**SaO** μ	$F - Statistic$ $p - value$
Number of Policy Templates				
Unconcerned	3.44	3.77	3.22	F(2,24)=0.34 $p = 0.71$
Pragmatist	3.44	4.69	4.47	F(2,186)=8.65 $p < 0.01$
Fundamen-talist	3.11	3.55	3.92	F(2,78)=2.89 $p = 0.06$
All	3.35	4.30	4.21	F(2,294)=9.71 $p < 0.01$
Policy Authoring Time (seconds)				
Unconcerned	213.0	144.0	160.7	F(2,24)=2.25 $p = 0.12$
Pragmatist	223.3	157.8	177.8	F(2,186)=8.52 $p < 0.01$
Fundamen-talist	227.7	148.2	152.7	F(2,78)=5.47 $p < 0.01$
All	223.6	153.9	169.4	F(2,294)=15.84 $p < 0.01$
Policy Openness (see Definition 1)				
Unconcerned	67.7	69.9	69.6	F(2,24)=0.01 $p = 0.98$
Pragmatist	67.8	56.8	53.2	F(2,186)=5.18 $p < 0.01$
Fundamen-talist	60.2	55.4	51.5	F(2,78)=1.24 $p = 0.29$
All	65.7	57.6	54.2	F(2,294)=5.13 $p < 0.01$

Same-As Subject and Same-As Object improve upon Subject / Object Grouping. We found that Grouping Based is more limiting in how a user may express their policies. On average, users of Grouping Based only leverage three policy templates versus four for Same-As Subject and Same-As Object. Fewer policy templates reflect that the user is authoring policies that aren't as expressive as they would like them to be. With Grouping Based, a user is being forced into expressing their policy in a way that may not align with their mental model – how the user views a policy versus how the access control model allows that policy to be expressed. We also see this reflected in Figure 5(a) where Grouping Based has a smaller distribution of policy templates versus Same-As Subject and Same-As Object.

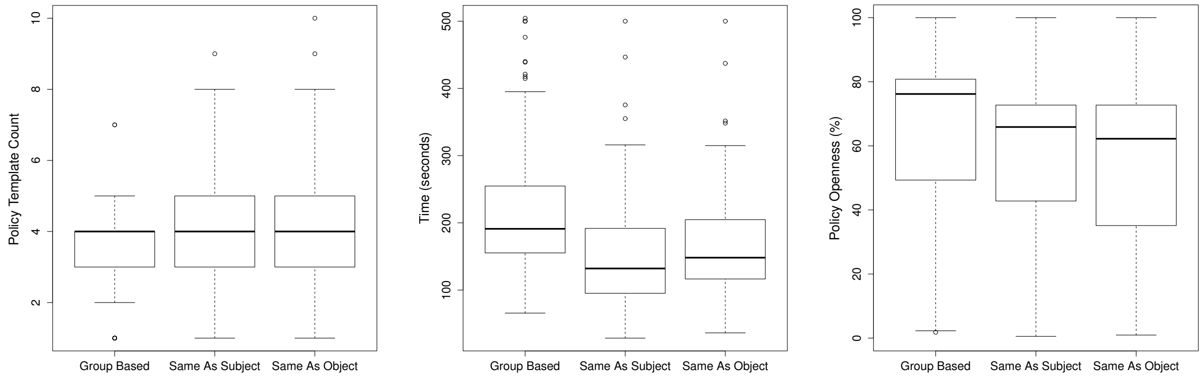
TABLE III. GROUPING BASED VS. SAME-AS SUBJECT VS. SAME-AS OBJECT – PAIRWISE COMPARISON

Measurement	**GB vs. SaS** $p - value$	**GB vs. SaO** $p - value$	**SaS vs. SaO** $p - value$
Policy Templates	< 0.01	< 0.01	1
Authoring Time	< 0.01	< 0.01	0.70
Flexibility	< 0.01	< 0.01	0.61
Readability	< 0.01	< 0.01	< 0.01
Openness	0.08	< 0.01	1
Ease of Use	0.07	< 0.01	0.53

b) Policy Authoring Time: Next, we set out to measure how long it took a user to author all their policies for each experiment. For Grouping Based, policy authoring time included grouping of friends, grouping of pictures and setting of permissions. For Same-As Subject, policy authoring time included grouping of pictures and setting of permissions for friends using the Same-As Example Subject as the policy template. For Same-As Object, policy authoring time included grouping of friends and setting of permissions for pictures using the Same-As Example Object as the policy template.

For Pragmatists, Fundamentalists and the population as a whole, we see statistical significance as it pertains to policy authoring time – all *p-values* are less than 0.01 and *F-Statistics* are greater than 3.04 (Pragmatist), 3.11 (Fundamentalist) and 3.02 (All) for a probability of 95%. Refer to the Policy Authoring Time section of Table II and Figure 5(b). We also ran a pairwise comparison leveraging the Bonferroni correction and observed no statistical significance between Same-As Subject and Same-As Object. However, we do see statistical significance between Grouping Based and Same-As Subject and Grouping Based and Same-As Object. See Table III. Pragmatists, Fundamentalists and the population as a whole, took less time authoring their policies with the Same-As approaches over the Grouping Based approach. Overall, we see a 31% reduction in policy authoring time when using Same-As Subject (153.9 seconds) versus Grouping Based (223.6 seconds). We also see a 24% reduction in policy authoring time when using Same-As Object (169.4 second) versus Grouping Based.

One factor attributing to this performance improvement is that with Grouping Based, a user must complete three disjoint tasks, i.e., group subjects, group objects and set permissions. With Same-As Subject and Same-As Object, a user first groups their objects or subjects, respectively. Then within one task, a user authors their policy by setting permissions using the Same-As Example policy template. With the Same-As approaches, the user is conducting fewer mental task switches. Conversely with Grouping Based, a user must focus on their relationship with their subjects and how they should be organized / grouped. Next, the user must think about their objects and how they are similar from a sensitivity perspective. Finally, the user must think about access permissions when they are authoring their policies. With the Same-As approaches, the user relies on their memory and opinion of their subjects or



(a) Number of Policy Templates

(b) Policy Authoring Time

(c) Policy Openness

Fig. 5. Grouping Based vs. Same-As Subject vs. Same-As Object

objects to set policies for other similar subjects or objects. As a result, users can author policies much faster.

c) Flexibility: Each user study particpiant completed a brief survey designed to capture their perceptions after each experiment. 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: Flexibility, Readability and Ease of Use.

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.

For Flexibility, we see statistical significance across all the user segments (minus Unconcerned). See Figure 6(a). We also ran a pairwise comparison and observed no statistical significance between Same-As Subject and Same-As Object. However, we do see statistical significance between Grouping Based and Same-As Subject and Grouping Based and Same-As Object. Refer back to Table III. Overall, users found Same-As Subject (5.43) and Same-As Object (5.65) more flexible than the Grouping Based (4.89).

Grouping Based provides fewer options for expressing one's policy. Same-As Subject and Same-As Object provide a means that aligns with how the user views their policies. Users are thinking about their friends or pictures when they are setting access permissions. With the Same-As approaches, users can create any number of different permutations for expressing their policy, all aligning with their intentions. This is reflected in our Flexibility measurement. In access control terms, the Same-As approaches have more expressive power for representing policies than does Grouping Based.

d) Readability: 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.

For Readability, we see statistical significance across all the user segments (minus Unconcerned). See Figure 6(b). We also see significance across all experiment pairings. See Table III. Overall, users found Same-As Object (5.71) more readable than Same-As Subject (5.20) than Grouping Based (4.64).

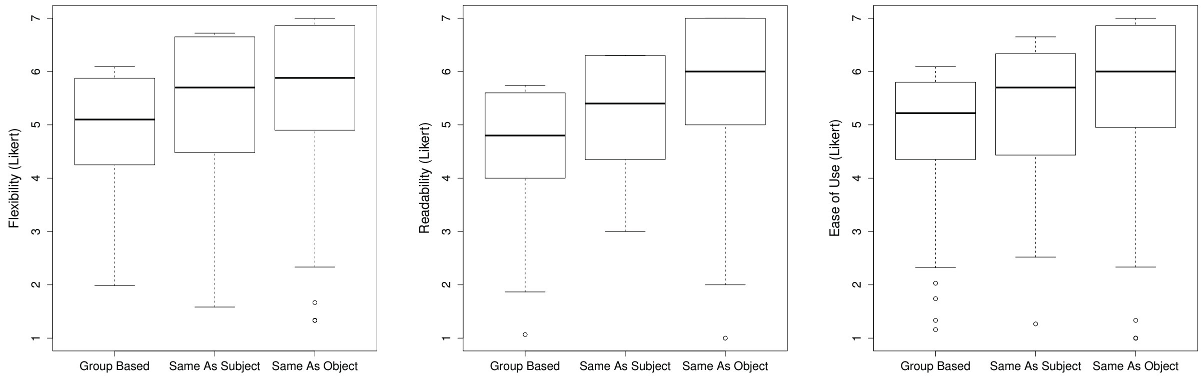
Using our visual policy editor, users are able to see the summarized expressiveness of their policy in a format this is easy to understand. With Grouping Based, a user must change between the different grouping and policy views to get a comprehensive understanding of their policy. Our visual policy editor presents the policy in a single view providing a global perspective to the user which is decipherable and aligns with their mental model. This allows the user to construct policies that align with their intent.

Readability is the first measurement where we see Same-As Object not only outperforming Grouping Based but also outperforming Same-As Subject. We attribute this to the introduction of our new paradigm in how objects are managed with Same-As Object. In both Grouping Based and Same-As Subject, objects are grouped in the traditional fashion, i.e., by some common property – in our study, sensitivity level. Using Same-As Object, the user associates objects with other like objects that possess a common user assigned subjective meaning of sensitivity. The user assigns the importance of the object by associating it with an easy to remember representative object, thus creating a level of abstraction based on the user's intent. In doing so, the policy is more readable and understandable to the user.

e) Policy Openness: We examined the *openness* of each user's policy or conversely, the conservativeness of a user's policy. This measurement gives an indicator of the restrictiveness (or not) of a user's policy, where a measurement of 100% indicates a totally permissive policy and a measurement of 0% indicates that the policy provides no access. We define Policy Openness as follows:

Definition 1: (Policy Openness) The probability of a user permitting a friend access to an 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 an object o and F_u is the overall friend set of u.

For Fundamentalists, there is no statistical significance as it pertains to Policy Openness ($p = 0.29$). See the



(a) Flexibility

(b) Readability

(c) Ease of Use

Fig. 6. Grouping Based vs. Same-As Subject vs. Same-As Object

Policy Openness section of Table II and Figure 5(c). There is statistical significance for Pragmatists and the population as a whole; both *p-values* are less than 0.01. In running a pairwise comparison, the only observed statistical significance is between Grouping Based and Same-As Object. See Table III. Same-As Object policies were more conservative than Grouping Based policies, 54.2% versus 65.7%.

We see Same-As Object policies to be more conservative (less open or permissive) than those policies authored using Grouping Based. Users are thinking about their objects in terms in which they assign, which aligns with their mental model and intentions. Therefore, they are more likely to create more expressive policies which are more *least-privilege* like, i.e., conservative. Flexibility and Readability also contribute to the conservativeness of user's policy.

f) Ease of Use: The user needs to be able to manage their access control 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. We see statistical significance across all the user segments (minus Unconcerned) for Ease of Use. See Figure 6(c). However, in doing a pairwise comparison, we only found statistical significance between Grouping Based and Same-As Object. Refer back to Table III. Overall, users found Same-As Object easier to use than Grouping Based – 5.59 versus 4.92 on a seven point Likert-scale.

Users found the Same-As Object approach and visual policy editor simple to use and easy to convey their access control intentions. Readability and Flexibility, in addition to reduced policy authoring times, are all contributors to the improved Ease of Use results for Same-As Object over Grouping Based.

VI. Conclusion

We introduce two usable object management approaches for online social networks. We extend our previous work to allow for object grouping. In addition, we introduce a new approach for managing objects for access control purposes, called Same-As Object Management. Same-As Object Management demonstrated to be more effective, efficient and satisfying to the user over more traditional object management approaches. Same-As Object Management leverages a user's memory and perception of their objects for setting permissions for other similar objects. This approach demonstrated to be more flexible; users were able to author more expressive policies that aligned with their mental model and intentions. These policies are also more readable than ones created using more traditional grouping based approaches. Because of the improved expressiveness and readability, policies are more conservative (less permissive) resulting in better security. Also, policy authoring time is reduced and users perceived Same-As Object Management easier to use.

References

[1] C. Birge. Enhancing research into usable privacy and security. In *Special Interest Group on Design of Communication*, 2009.

[2] G. P. Cheek and M. Shehab. Policy-by-example for online social networks. In *Proceedings of the Symposium on Access Control Models and Technologies*, 2012.

[3] S. Egelman, A. Oates, and S. Krishnamurthi. Oops, i did it again: mitigating repeated access control errors on facebook. In *Conference on Human Factors in Computing Systems*, 2011.

[4] D. Ferraiolo and R. Kuhn. Role-based access control. In *Proceedings of the National Computer Security Conference*, 1992.

[5] N. A. Jones, H. Ross, T. Lynam, P. Perez, and A. Leitch. Mental models: An interdisciplinary synthesis of theory and methods. In *Ecology and Society*, 2011.

[6] S. Jones and E. O'Neill. Feasibility of structural network clustering for group-based privacy control in social networks. In *Symposium on Usable Privacy and Security*, 2010.

[7] H. Krasnova, O. Gnther, S. Spiekermann, and K. Koroleva. Privacy concerns and identity in online social networks. *Identity in the Information Society*, 2009.

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

[9] B. Lampson. Privacy and security: Usable security: how to get it. *Communications of the ACM*, 2009.

[10] 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*, 2008.

[11] M. Moyer and M. Abamad. Generalized role-based access control. In *The International Conference on Distributed Computing Systems*, 2001.

[12] R. W. Reeder, L. Bauer, L. F. Cranor, M. K. Reiter, and K. Vaniea. More than skin deep: measuring effects of the underlying model on access-control system usability. In *Proceedings of the Conference on Human Factors in Computing Systems*, 2011.

[13] J. Saltzer and M. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 1975.

[14] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 1996.

[15] R. Sandhu, D. Ferraiolo, and R. Kuhn. The nist model for role-based access control: Towards a unified standard. In *Proceedings of the ACM Workshop on Role-Based Access Control*, 2000.

[16] M. Theofanos and S. Pfleeger. Guest editors' introduction: Shouldn't all security be usable? *IEEE Security & Privacy*, 2011.