Automatic Social Group Organization and Privacy Management

Anna Squicciarini
Information Sciences & Technology
Pennsylvania State University
Email: asquicciarini@ist.psu.edu

Sushama Karumanchi
Information Sciences & Technology
Pennsylvania State University
Email: sik5273@ist.psu.edu

Dan Lin
Department of Computer Science
Missouri University of Science & Technology
Email: lindan@mst.edu

Nicole DeSisto
Information Sciences & Technology
Pennsylvania State University
Email: ndesi2@brockport.edu

***Abstract*—With the dramatic increase of users on social network websites, the needs to assist users to manage their large number of contacts as well as providing privacy protection become more and more evident. Unfortunately, limited tools are available to address such needs and reduce users' workload on managing their social relationships. To tackle this issue, we propose an approach to facilitate online social network users to group their contacts into social circles with common interests. Further, we leverage the social group practice to automate the privacy setting process for users who add new contacts or upload new data items. We conducted a user study to evaluate the effectiveness of our solution.**

I. Introduction

Social networking sites are proliferating fast with an increasing number of users and increasingly complicated social relationships among users. Micro-managing this large amount of personal data has shown to be a very burdensome task for regular users, as acknowledged by a growing number of research studies and news articles [1], [4], [9], [12], [17], [22]. It is even more challenging to configure proper privacy settings for data being shared in social networking sites. Security unaware users typically follow an open and permissive default policy. As a result, the potential for unwanted information leakage is great.

To tackle the above problems, we introduce an approach to facilitate the users to management their social relationships as social groups, and then we leverage the social groups to provide privacy setting recommendation for users. Our approach builds on the following rationale. As confirmed by the most recent social network platforms, social circles in modern social networks can act as the foundation of user management and privacy management. For instance, Facebook provides an optional mechanism that allows users to create custom lists to organize friends and set privacy restrictions accordingly. Facebook also recently announced smart lists which automatically group friends who live near by or attend the same school. Similarly, the newly released Google+ creates four default circles for users: Friends, Family, Acquaintances, and Following. A user can remove/rename any of the default circles or add new circles. For privacy management, users in Google+ can selectively share information with a specific set of their circles, all their circles, their extended circles or with the public (everyone).

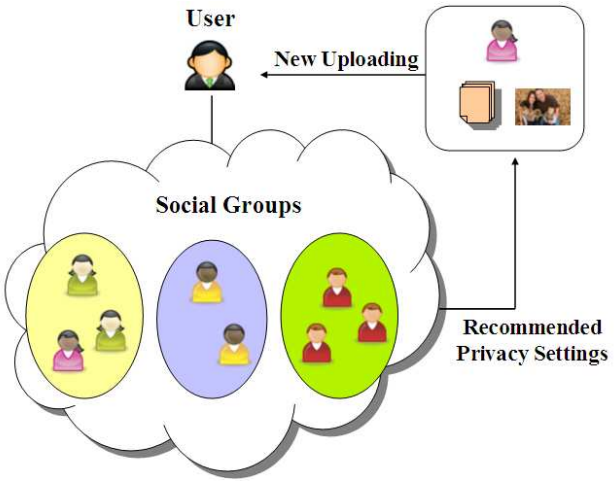

Fig. 1. Policy Recommendation Using Social Groups

While the idea of social circles is very interesting and promising, existing social network platforms have not fully explored the full benefit of this concept and their related systems are at primitive stage with no or limited support on circle formation and privacy management. As an advance in this direction, we design a multi-criteria model that takes into account multiple aspects of users' profiles, and automatically groups each user's contacts into social circles with common characteristics. Users in the same social circle (group) have similar behavior, such as similar education background, hobbies, and similar privacy preferences. Given the obtained grouping information, we further propose an approach to

recommend privacy policies for newly uploaded data items or newly added contacts. In particular, when a user uploads an object (a data item or a contact), our system looks for the social group which is most likely to deal with the object in the similar way as the user, and then the privacy settings adopted by the selected group is considered as the base for predicting policies for the new object. Figure 1 gives an overview of our approach.

Our approach has the ability to identify hidden groups which may play an important role during privacy settings, but may not be identified by users. For example, within a user's friend list, there may exist a sub-group which includes mainly close friends with whom the user shares a large amount of data; in a user's family group, there may be direct family members with whom the user shares family pictures, events (e.g., anniversary, notes). Privacy settings are likely to be different in each such hidden sub-group, and hence identifying these sub-groups will help enhance and simplify users' privacy practices. Our approach has been verified in terms of effectiveness via a user study.

The rest of the paper is organized as follows. Section II discusses related works with respect to social circles and associated privacy management issues. Section III introduces notations and defines the problem. Section IV presents the detailed algorithms for social grouping, followed by Section V which leverages the grouping information for policy prediction. Then, Section VI reports experimental results. Finally, Section VII concludes the paper and outlines future research directions.

II. Related Work

Work on social networking privacy enhancing technologies is nowadays proliferating. In particular, several recent works have studied how to automate the task of privacy settings [3], [5], [6], [10], [18], [20].

Bonneau et al. [5] proposed the concept of privacy suites for social network sites, based on the idea that most users currently stick with default privacy settings. In particular, they recommend to users a suite of privacy settings that expert users or other trusted friends have already set, so that normal users can either directly choose a setting or only need to do minor modification to available settings. Along similar lines, Danezis [6] proposed a machine-learning based approach to automatically extract privacy settings from the social context within which the data is produced. Parallel to the work of Danezis, Adu-Oppong et al. [10] develop privacy settings based on a concept of social Circles which consist of clusters of friends formed by partitioning users friend lists. Ravichandran et al. [20] studied how to predict a user privacy preferences for location-based data (i.e., share her location or not) based on location and time of day. Fang et al. [11] proposed a privacy wizard to help users grant privileges to their friends. The wizard asks users to first assign privacy labels to selected friends, and then uses this as input to construct a classifier which classifies friends based on their profiles and automatically assign privacy labels to the unlabeled friends. Subsequently, the same research group [3] introduced a policy visualization tool which displays privacy settings for user specific subgroups of friends within social networks. Liu and Terzi [16] have defined a mathematically sound methodology for computing users privacy scores in online social networks. The privacy score indicates the users potential risk caused by his or her participation in the network. The authors definition of privacy score satisfies the following intuitive properties: the more sensitive information a user discloses, the higher his or her privacy risk.

Jones et al. [15] investigate users' rationales for grouping friends, for privacy management purposes, within online social networks. They identify six static criteria for grouping, and evaluate the similarity of these criteria to the output of standard clustering techniques of users' friends. Their work supports our notion that standard clustering techniques can assist users in placing friends into groups analogous with privacy intentions.

Finally, Hu and colleagues [13] have studied data sharing in social networks, with emphasis on conflict resolution in case of multiple party involved, similar to [8].

III. Problem Statement

In this section, we introduce notations and definitions adopted in this paper.

Social networks are qualified by a set of users, a set of user profiles, a set of user contents, and a set of user relationships. A user profile indicates who a user is in the social network, such as their identity and personal information. User content describes what a user has exposed in the social network, such as uploaded photos, videos, blogs, and other data objects created through various activities in the social network. User relationships represent user connections with friends, family, coworkers, colleagues, etc.

Formally, the social network is defined as follows.

Definition 1: (Social Network) A SN is a labeled graph $\langle U, E, \Phi \rangle$, where U denotes the set of nodes and E the labeled edges. Each node represents a user u_i. Each edge $E_{i,j}$ represents a relationship between users u_i and u_j, where u_i and u_j are unique identifiers of users. Edges are labeled with the social relationship type that connects the two users. The labeling function Φ is defined as $\Phi : U \times U \rightarrow \mathcal{P}(\mathcal{R})$, where U is the set of users registered to the SN and $\mathcal{R} = \{R_1, \ldots, R_m\}$ is the finite set of the possible relationships connecting the users. A relationship R_k connecting users i and u_j is denoted as (u_i:R_k:u_j). The relationship R_k is bidirectional, therefore (u_i:R_k:u_j)=(u_j:R_k:u_i).

Each user $u_i \in U$ is represented as a vector $prof_i$ in the form of $[P_1(i), ..., P_w(i)]$, where $P_k(i)$ is u_i's k-th property ($1 \leq k \leq w$). Properties are sorted by time of creation. A property is represented as a pair: $P_k(i) = (pn_k(i), pv_k(i))$, where pn_k is the property name, and pv_k is the property value. Some properties may have a single unique value, such as one's home town, whereas some properties may have multiple

values, such as the schools attended or hobbies. For a multi-valued property, we store multiple pairs corresponding to each value. For example, a user who enters jogging and swimming as his favorite activities will have his property namely "favorite_activity" stored as two pairs: (favorite_activity, jogging) and (favorite_activity, swimming).

In our system, we support properties of several types (PTypes): (1) regular attributes (PT_{attr}); (2) users' relationships (PT_{rel}); (3) images (PT_{image}); (4) comments and posts (PT_{comm}); (5) privacy preferences (PP); (6) social groups (PT_{mm}). We elaborate on each of the property type in the following:

- PT_{attr}: denotes attributes that have text or number values and are typically used to describe the user. For example, regular attributes include the user's name, gender, birth date, occupation, affiliation, address, hobbies, education background, etc.
- PT_{rel}: indicates relationship between the user and his/her contacts. The name of this property is "Rel", and its value is in the form of u_i:*R*:u_j as defined in Definition 1. For example, if user u_1 has a friend u_2, u_1's PT_{rel} is represented as ("Relationship", u_1:friend:u_2). If user u_1 has more than one friend, a pair of property name and value is created for each of u_1's friend. Such representation is for the ease of the user grouping introduced in the next section.
- PT_{image}: denotes the image file uploaded by the user u_i, and is described by two concatenated strings $u_i : pid$, where pid is the unique identifier of the image.
- PT_{comm}: represents streaming data, that is, blogs, posts, comments, and other texts that users post on each other's profile. The name of this property is "Comment", and its value is the text file input by the user. For example, user u_i uploads a blog file $vacation.txt$, which is represented as ("Comment", vacation.txt).
- PP: records the privacy policies specified by a user. The name of the property is "Policy", and the value of this property is a privacy policy in the form of $\langle \texttt{R}, \texttt{objt}, \texttt{cond}, \texttt{priv} \rangle$, where R refers to the relationship type the policy applies to (i.e., friends), $objt$ refers to the type of object being protected (i.e., images, text, portion of profile), $priv$ is the privilege, and $cond$ is Boolean expression that defines the constraints under which the $priv$ is granted. An example privacy policy is given below, which means Bob is allowed to comments on the policy owner's friend photos and blogs at anytime:
  P_1: $\langle$Bob, {friend_photos, myblog}, comments, anytime$\rangle$.
- PT_{mm}: is used to model the social group membership of a user. The property name is "Group" and the value is equal to the name of the group joined by the user. For example, when the user Jane joins the group "Fashionista", a new property ("Group","Fashionista") is inserted to Jane's property vector.

Our problem is twofold. The first problem is to automatic group a user's contacts into social groups to ease the management burden opposed on users, the details of which are introduced in Section IV. The second problem is to utilize the identified groups to help users specify best privacy policies for a newly uploaded data item or a newly added contact (Section V).

IV. Finding Social Groups

As aforementioned, one of our goals is to alleviate users' burden on managing their social relationships by automatically grouping a user's contacts into social groups with common characteristics. One intuitive approach to identify groups in an online community is to group users based on a pre-selected property. For example, if one selects the "hobby" as the grouping criteria, social groups of same hobbies will be generated. More specifically, according to the value of the property "hobby", users who like "fishing" will be placed in the same group while users who like "hiking" will be placed in another group. Here, the common characteristics being considered during the grouping is the arbitrarily selected property. The selection of the grouping property directly affects the quality of the grouping whereas the selection is not a trivial task for the user. Moreover, such approach may not be able to truly capture the similarity among contacts with respect to a user. Consider the following example.

Example 1: Suppose that user u_1 has six contacts u_2, u_3, u_4, u_5, u_6, and u_7. For simplicity of illustration, consider that each user has only three properties.

u_1: [(education, "PennState"), (hobbies, "swimming"), (rel, 1:friend:2)]
u_2: [(education, "PennState"), (hobbies, "hiking"), (rel, 2:friend:1)]
u_3: [(education, "PennState"), (hobbies, "tennis"), (rel, 3:friend:1)]
u_4: [(education, "Stanford"), (hobbies, "basketball"), (rel, 4:friend:1)]
u_5: [(education, "UCLA"), (hobbies, "PCgame"), (rel, 5:friend:1)]
u_6: [(age, 50), (hobbies, "movie"), (rel, 6:family:1)]
u_7: [(age, 53), (hobbies, "movie"), (rel, 7:family:1)]

If user u_1 randomly selects a property (e.g., "hobbies") as the grouping criteria, the results will contain five groups since all his contacts have different hobbies. In fact, a more natural grouping could be:

Group 1: (u_2, u_3), they are probably schoolmates of u_1.
Group 2: (u_4, u_5), they are other friends of u_1.
Group 3: (u_6, u_7), they are probably u_1's parents.

Observe that Group 1 is obtained based on two properties: education and relationship. Group 2 is obtained based on the relationship property, and Group 3 is obtained based on age, hobbies and relationship. Such grouping does not rely on a fixed property, and it captures the similarity among u_1's contacts much better than the one using the pre-selected property. Further, it may later be used to derive useful information about users' privacy preferences in the system.

To achieve the effect of the grouping as discussed in the above example, we need an approach to dynamically detect grouping criteria, i.e., certain combinations of properties. Thus, we propose a modified version of the data mining algorithm based on the apriori algorithm [2], to extract frequent features (i.e., frequent occurring combinations of properties) of a user's contacts, and then design the algorithm to carefully

select features for grouping. In what follows, we first introduce how to represent and compare features pertaining to different users and then present the detailed grouping algorithm.

A. Feature Representation and Matching

The base of the user grouping is the feature mining which aims to identify frequently occurring combinations of properties. However, many of the user properties have a complex and heterogenous structure. In order to conduct an effective feature mining, we need a comparison algorithm to determine the similarity among the various types of properties.

The exact list of properties to be compared may be domain dependent. As a preparation step, we filter out properties that do not have significant impact on the users' privacy preferences, such as user names. Considering that users post on average about 90 pieces of content per month on popular social networks [7] and the property vector would likely explode, we only keep properties that have shown to be relevant for describing the users' social capital in online communities, and that are significative descriptives of the users' social activities. Take the Facebook as an example, we consider the following properties during the feature extraction: relationship, location, hobbies, age, and privacy preference.

As for property comparison, a straightforward thought is to employ exact mapping on same type of properties. However, this is not effective for the purpose of this work, i.e., to capture similarity among users described by these properties. This is because some of the original properties of the user are not informative when considered individually. For example, social contacts are not significant features, when individually considered, but bear some weight when they are indicative of a significant overlap of users' social graphs. That is, if user u_i and u_j have properties of type relationship (i:$Friend$:m) and (j:$Friend$:m), respectively, the common friend m represents an interesting correlation between the two users. Further, it is unlikely that the single common friend would be of any significance. The fact that user u_i and u_j share many friends is more relevant. Therefore, in order to provide a better indication of a specific characteristic (i.e. feature) of a user, we aggregate all the properties of relationship and consider them as a group. Similarly, the differences in terms of privacy preferences should be represented as general privacy preferences, to denote the overall users' preferences.

After property aggregation, we then conduct the discretized matching of properties. Below we discuss how different types of property is considered and matched.

- *Relationship*: Two users u_i and u_j have a matching feature for a given relationship *R* if the profiles $prof_i, prof_j$ include properties $(i : R : k), (i : R : k), (i$:$R$:$t), \ldots, (i$:$R$:$z)$ and profile $prof_j$ includes $(j : R : k'), (i : R : t'), \ldots, (i : R : z')$ where $R = R'$ and $k = k', z = z'$ for at least 30% of the PT_{rel} of type R in u_i and u_j.
- *Location*: We measure locations latitudes and longitudes distance, and consider two locations to be the same if they are within a certain geographical proximity. Specifically, given two profiles $prof_i, prof_j$, the location of i, j is a common feature if there is a property in i, j profiles denoting two locations A and B, respectively, and A and B are in their geographical proximity.
- *Hobbies*: We account for the possibility of users expressing interests in different way. For example a user may indicate "Running" while another one may indicate "Jogging". To account for the syntactic differences, we use the Wordnet classification structure. Precisely, $(Hobby, X), \ (Hobby, Y)$ are matches if Y, X share a hypernym.
- *Age*: We represent age using discrete ranges or categories, e.g. $[< 18], [28, 25]$.... Two users i, j have matching age if (Age, X) and (Age, Y) and X, Y belong to the same interval.
- *Privacy*: As comparing detailed privacy policies for every pair of users would be time consuming, we convert policies into strictness levels using the approach proposed in [21]. The strictness level is a quantitative metric that describes how strict a policy is. For example, a policy that allows only family member to download images is more restricted than a policy which allows any stranger to download the images. The value of the strictness level starts from 0. The lower the value, the higher the strictness level. The conversion from policies to strictness levels are conducted once and the values of the strictness levels are stored along with the policy. After conversion, we just need to compare the strictness levels of the corresponding policies during the feature mining.

B. Grouping Users

We aim to group a user's contacts into social groups so that each such social group shares common values for a certain set of properties. To formalize the problem, we first introduce the following definitions.

Definition 2: (k-group) Let U be the set of all users, and F be the universe of properties of users' profiles, F=$\{p_1, p_2, ...\}$. Let C be a set of k properties, C =$\{p_{i1}, p_{i2}, ..., p_{ik}\}$, where $p_{ij} \in F$. Let G be a subset of users U, i.e., $G \subseteq U$. G is a k-group if users in G has matching values for each property listed in C.

Example 2: Reconsider the five users in Example 1. Users u_1, u_2 and u_3 has two matching properties: education and relationship. Thus, G_1=$\{u_1, u_2, u_3\}$ is a 2-group where C_1=$\{$"education", "relationship"$\}$. Users u_6 and u_7 has three properties with matching values, and hence G_2=$\{u_6, u_7\}$ is a 3-group where C_2=$\{$"age", "hobbies", "relationship"$\}$.

To be of interest, we consider the k-group with more than certain number of users, and define such groups as frequent group in Definition 3.

Definition 3: (Frequent Group). Let G be the k-group with common properties in C. G is a frequent k-group if the number of users in G is no less than a threshold, i.e., $|G| \geq min\_sup$, where $min\_sup > 0$.

Example 3: Given $min\_sup = 3$, G_1 in Example 2 is a frequent 2-group since $|G_1| = 3 \geq min\_sup$; while G_2 is not a frequent group since $|G_2| = 2 < min\_sup$.

The problem of finding social groups for a user is now converted into finding all frequent k-groups for a user. To solve the problem, we employ the well-known data mining algorithm, Aprior algorithm [2] as follows. Recall that Apriori was originally designed to extract frequent patterns from transaction data. The Apriori algorithm takes a set of transactions as input and produces a list of frequent item sets. In our context, transactions are corresponding to users, items are corresponding to user properties, and item sets are corresponding to sets of properties.

The algorithm is a level-wise iterative search algorithm that uses the frequent k-communities to explore the frequent $(k+1)$-communities. Two frequent $k-1$-communities can be joined together to form a candidate k-community only if their first $(k-2)$ items match and their $(k-1)^{th}$ items are different. This operation is based on the Apriori property: A community cannot be frequent if any of its subsets is not frequent. Thus, the only potential frequent communities of size k are those that are formulated by joining frequent $(k-1)$-communities. Specifically in our case, we first find the set of frequent 1-group by scanning the users' profiles, accumulating the support count of each shared property, and collecting the groups with supports no less than min_sup. Then, we join every pair of frequent 1-groups and keep the non-empty joining result. Next, we join every pair of frequent 2-groups which have at least one property in common. Similarly the identified frequent 2-groups are used to find frequent 3-groups, and so on, until no more frequent k-groups can be found. During the process, if a frequent i-group cannot be used to produce $(i+1)$-group, this i-group is included in the final results.

Example 4: We consider users in Example 1 to illustrate the process of finding frequent k-groups. Given the min_sup=2, we find two frequent 1-groups:

$G_1 = \{u_1, u_2, u_3\}$, C_1 = {*education*}
$G_2 = \{u_1, u_2, u_3\}$, C_2 = {*rel*}
$G_3 = \{u_6, u_7\}$, C_3 = {*age*}
$G_4 = \{u_6, u_7\}$, C_4 = {*hobbies*}
$G_5 = \{u_6, u_7\}$, C_5 = {*rel*}

By joining frequent 1-groups, we obtain frequent 2-groups as follows:

$G_{1-2} = G_1 \bigcap G_2 = \{u_1, u_2, u_3\}$, C_{1-2}={*education, rel*}
$G_{3-4} = G_3 \bigcap G_4 = \{u_6, u_7\}$, C_{3-4}={*age, hobbies*}
$G_{3-5} = G_3 \bigcap G_5 = \{u_6, u_7\}$, C_{3-5}={*age, rel*}
$G_{4-5} = G_4 \bigcap G_5 = \{u_6, u_7\}$, C_{4-5}={*hobbies, rel*}

Next, we join frequent 2-groups. G_{1-2} cannot be joined with any other 2-groups to produce a 3-group, and thus G_{1-2} is included in the final result. The joining results of any two of G_{3-4}, G_{3-5} and G_{4-5} are the same, and hence we just need to keep one as follows.

$G_{3-4-5} = G_{3-4} \bigcap G_{3-5} = \{u_6, u_7\}$,
C_{3-4-5}={*age, hobbies, rel*}

To sum up, the final results contain one frequent 2-group, i.e., G_{1-2}, and one frequent 3-group, G_{3-4-5}.

In the process of the grouping, each group also maintains a summary profile to store the support of each property and the number of each type of data items uploaded by users. Upon time, if the change of the summary structure is greater than certain threshold after several rounds of updates of new contacts or new data items, consider the splitting of the group as well as merging with other groups with similar features. The change of the threshold is determined according to the support of features. If the support of the frequent features lose the dominant status, the group needs to be reconstructed.

V. Policy Prediction

This phase is to leverage the grouping structure to facilitate users to set appropriate privacy preferences. We consider two sharing problems: (1) appropriate privacy settings for a new contact added by a user u_i; (2) appropriate privacy settings for a given data item being added by the user u_i.

To solve the problems, the basic idea is to identify the most similar data item/contact in existing groups and then customize their policies for the new data item/contact. The intuition here is that a user typically has similar privacy concerns regarding similar data items (or contacts with similar properties). For example, family photos may usually be shared within the family members; blogs about working progress may usually be shared among colleagues in the same project team.

Given a user u_i who added an object (either a new data item or a new contact) O_i, the policy prediction algorithm conducts the following three phases: (1) determine the overall search scope; (2) Locate objects similar to the newly added object O_i; (3) generate the privacy policy.

The first phase aims to narrow the search range from the entire social network to a few social groups that are closely related to the user u_i and may contain objects similar to O_i. Here, we not only consider the user who uploaded the new object but also his/her closely related contacts in order to generate a wider yet still appropriate base for policy prediction. In particular, we will consider the social groups of users satisfying the following condition: if user u_j is u_i's contact, and u_j has more than N contacts in common with u_i, where N is set to 50% of u_i's contacts. The reason to consider groups other than the one pertaining to u_i is two-fold. First, users with many common contacts usually share many things in common (e.g., similar hobbies, similar background). More importantly, one of the grouping criteria is privacy concerns. Second, it provides a larger pool of data to facilitate the understanding of the privacy tendency and hence the privacy prediction can be more accurate.

The second phase is to compare the social groups returned by the first step and selects the one which is most likely to contain objects similar to O_i. The comparison algorithm differs according to the type of the uploaded object. If the uploaded object is a new contact, the properties of the new contact are compared against the social group features using

the distance function defined in Equation 1.

$$Diff(O_i, G_j) = \Sigma_{k=1}^{n} D(p_k^{o_i}, p_k^{G_j}) \quad (1)$$

In Equation 1, the distance between a new contact and the social group is measured as the total difference between each pair of corresponding properties. The social group with the smallest distance will be selected for the further consideration in the next step.

If the uploaded object O_i is a data item such as photos or blogs, we will find the social group which contains the largest number of data items similar to O_i. Specifically, we first check the summary structure of the social groups in the search range, and sort them in a descending order of the number of data items that have the same data type of O_i. Then we start to examine the social group from the top of the sorted list. Given the nature of the data item O_i, the corresponding image-content analysis tool [14] or text analysis tool [19] is utilized to compute the similarity between O_i and the data items in the examined social group. For each social group, we count the number of data items which have the similarity score above the average similarity score. Then the social group with the highest count is returned as the input of the third step.

Finally, the third phase is to analyze the privacy policies specified by the users in the selected social group. Recall that a policy consists of four components: $\langle$R, obj, right, condition$\rangle$. First, we filter out the policies that do not contain objects similar to O_i. In particular, if O_i is a new contact, O_i's properties will be compared with R's properties. If O_i is a data item, O_i will be compared with obj in the policy. If the similarity score is lower than certain threshold, the corresponding policy will not be further considered. Then, among the remaining policies, we execute the Aprior algorithm on the policy components excluding the one that the uploaded object belongs to. That is, the object component $objt$ in the policy will be removed from the pattern mining when the uploaded object is a data item; the R component will be removed when the uploaded object is a new contact. The mining results contain frequent patterns made of the combinations of the other three policy components. These frequent patterns will be customized to form complete policies as follows. For the pattern that contains all components except the one corresponding to the uploaded object, we will add the uploaded object to the pattern to form a policy. An example scenario is given below.

Example 5: Suppose that user u_i added a new friend O_i; there are 10 policies in the social group returned by the second phase. Among the 10 policies, only 5 policies are specified for friends as listed below:

P_1:⟨Bob, {friend_photos, myblog}, comments, anytime⟩.

P_2:⟨Alice, {friend_photos, family_photos, myblog},comments, anytime⟩.

P_3:⟨Tom, {friend_photos, myblog}, comments, anytime⟩.

P_4:⟨Kate, {friend_photos, myblog}, comments, anytime⟩.

P_5:⟨Jack, friend_photos, viewonly, 1/1/2012 − 1/1/2013⟩.

Excluding the subject, the most frequent pattern containing three components is: "{friend_photos, myblog}, comments, anytime". Based on this information, the system generates the following policy for O_i.

P_{oi}: ⟨O_i, {friend_photos, myblog}, comments, anytime⟩.

In case the identified frequent patterns do not contain all three components, we sort them in a descending order of the support. Then, we select the missing components from the sorted list to form one complete policy. The policy formed using such combination of frequent patterns will be marked since it may not be the most accurate one. while it may give the user hints what are the popular actions, conditions that being used. Also, the final output may contain multiple policies.

An example of policy prediction using the combination of frequent patterns is given below.

Example 6: Suppose that the policies in the Example 5 are modified as follows:

P_1: ⟨Alice, {friend_photos, family_photos, myblog}, download, anytime ⟩.

P_2: ⟨Bob, {friend_photos, myblog}, comments, 1/1/12-1/1/13⟩.

P_3: ⟨Tom, {friend_photos, myblog}, comments, 1/1/12-1/1/13⟩.

P_4: ⟨Kate, {friend_photos, myblog}, comments, anytime⟩.

P_5: ⟨Jack, friend_photos, viewonly, 1/1/12 − 1/1/13⟩.

Excluding the subject, the frequent patterns are:

"{friend_photos, myblog}, comments", support = 3.

"1/1/12-1/1/13", support = 3.

The policy P_{oi} is the result of the combination of these two frequent patterns.

P_{oi}: ⟨O_i, {friend_photos, myblog}, comments, 1/1/12-1/1/13⟩.

Figure 2 outlines the policy prediction algorithm. Lines 1–4 find the set of social groups (denoted as SG) related to the user u. Lines 5–10 identify the social group which is

Algorithm: Policy_Prediction(u, O)

Input: O is an object uploaded by user u

1. $SG \leftarrow \emptyset$
2. For each user u_i in social network
3. If u_i and u_j have more than %50 common contacts
4. $SG \leftarrow SG \bigcup SG_{u_i}$
5. MinDiff←0
6. For each G_i in SG
7. Diff← Difference(G_i, O)
8. If Diff<MinDiff
9. BestG← G_i
10. MinDiff←Diff
11. TranSet← $\emptyset$
12. For each policy P in BestG
13. If P contains R or obj similar to O
14. Tran← (P excluding component similar to O)
15. TranSet← TranSet $\bigcup$ Tran
16. FPset← Apriori(TranSet)
17. FP3set← frequent patterns with 3 components in FPset
18. If FP3set is not empty
19. Find the frequent pattern FP in FP3set with highest support
20. Generate the predicted policy based on FP
21. Else
22. Sort patterns in FPset in a descending order of support
23. Combine patterns in FPset to form the predicted policy

End Algorithm.

Fig. 2. Policy Prediction Algorithm

most likely to contain the objects similar to the object O uploaded by u, which is denoted as $BestG$. Lines 11–16 select policies containing objects similar to O and run the frequent pattern mining algorithm, the Apriori algorithm on the policy components. Finally, lines 17–23 assemble the frequent patterns to form the predicted policy.

VI. Performance Study

We have carried out a user study to collect real-world data and evaluate our proposed approach. In what follows, we first introduce the set up of the user study, and then present our findings.

A. Experimental Settings

Our user study collects users' profiles and asks users what policies they would have for given scenarios. The collected real policies are then used as ground truth to compare with our predicted policies. In particular, we recruited 140 undergraduates from the same undergraduate course at the Pennsylvania State University. There are 62% female and 38% male, and the average age is 21 (std. 0.4). The participants were asked to complete a questionnaire consisting of two parts:

- Questionnaire Part I: This part includes questions that ask about users' demographics and habits in social networking sites, such as their interests, privacy settings, types of content typically posted, amount of content. We also asked the participants to indicate their top contacts (up to 30) within the class wherein we conducted the survey, so as to simulate social relationships.
- Questionnaire Part II: This part introduces 15 distinct scenarios as summarized in Table VI-A. The scenarios cover three types of contents: images (in Scenario 1 to 5), text (in Scenario 6 to 13), and video (in Scenario 14 and 15). These scenarios simulate situations when the uploaded objects may be associated with different levels of privacy concerns. For example, family photos (Scenario 1) are generally more private than scenery photos (Scenario 3).

  For each scenario, the participants need to answer the following questions on privacy preference settings: (1)"Who would you like to share the photo with and for how long?"; (2)"What permissions would you like to give to them?". To assist the participants to answer the questions, we provide a set of options reproducing a classic access control policy of a social networking site. Each policy has six conditions, related to the access privilege being granted, e.g., view, re-share, comment, tag, and some additional temporal constraints, wherein participants can choose whether to grant limited or permanent access. The participant is asked to provide 6 policies for each scenario, targeting six different demographics (e.g., friends, close friends, acquaintances, etc.). Each allowed the participants to indicate the access mode (e.g., view, comment, re-share) and the temporal component (e.g., indeterminate, temporary). This has allowed us to obtain a ground-truth dataset of over 12,000 access rules (one access rule per relationship was obtained), which reproduces, although with some limitations, the rules users would put in real-world social networks.

TABLE I
Summary of Scenarios in Questionnaire Part II

No.	**Summary of Scenarios**
1	post a family photo on social network
2	post your own photo which shows you in a public place
3	post a scenery photo you took on a trip
4	post an interesting image that you found on the Internet
5	post a photo about your business meeting
6	post a blog about your weekend activities
7	post a blog about your hobbies
8	post a blog about your view of society issues
9	post comments about a movie
10	post a question to ask for help
11	post a note to look for new friends
12	post news that you found online
13	post news of your upcoming party
14	post an educational video (e.g., change tire)
15	post a family video

B. Experimental Results

From Questionnaire Part I, we noticed that 96.4% of the participants have an account in a social network site, and therefore the collected data is representative of active users in social sites. Regarding the privacy setting configuration, 57.7% of the participants have a customized setting which means the setting has been changed by the participants, 38.5% of the participants have the default setting configured by the social network, and 3.8% of the participants have a strict privacy setting, that is, only the participant himself/herself can see the content.

To start the policy prediction, we randomly select 80 policies from the actual policy set as the initial training dataset. For our evaluation, we primarily tested the accuracy of the predicted policies. We compare each corresponding pair of the predicted policy and the actual policy input by the participant. We count the number of mismatches in all the policy components, and measure the accuracy using the following error rate function.

$$Err(P_{pred}, P_{act}) = \frac{N_{err}}{max(N_{P_{act}}, N_{P_{pred}})} \quad (2)$$

In Equation 2, N_{err} is the total number of mismatching values in policy P_{pred} and P_{act}, and $N_{P_{act}}$ and $N_{P_{pred}}$ are the total number of values in the actual policy and the predicted policy respectively. Consider the following two example policies P_{act} and P_{pred}:

P_{act}: $\langle$`Kate`, {`photos`, **videos**}, `viewonly`, `anytime`$\rangle$.

P_{pred}: $\langle$`Kate`, {`photos`}, **comments**, `anytime`$\rangle$.

Observe that the predicted policy P_{pred} differs from the actual policy P_{act} in two places as highlighted in bold, i.e., N_{err}=2; there are four items (one item per policy component) in P_{act}, i.e., $N_{P_{act}} = 4$. Thus, the error rate is computed as $Err(P_{pred}, P_{act})$ = 2/5= 40%.

TABLE II
PREDICTION ERROR RATE

Scenario	Error Rate
1	27.5%
2	21%
3	30%
4	30.4%
5	25.5%
6	24.5%
7	28%
8	21%
9	30%
10	28.5%
11	22.5%
12	25.5%
13	3%
14	25%
15	27.5%

In the experiments, the average error rate for the 12,000 policies obtained from the 15 scenarios is about 24%. The error rates categorized by each scenario are shown in Table VI-B. Consider that a couple of mismatches can result in error rate as high as 40% as shown in the example. 25% error rate from the experimental results means that our policy prediction method is quite accurate.

When taking a further look at the mismatching values, we notice that the following policy conditions gave the highest number of errors as they are missing in the predicted policies: "Close Friend View", "Colleague Permanent Access", and "Close Friend Comment". That is, the predicted policy would not allow view and comment actions to close friends of the participant, or permanent access of a content to the colleagues of the participant. We argue that even though this is an error, it may be better to suggest a policy that is more restrictive than users expectation. In addition, users are always allowed to revise the predicted policies before real use.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed an approach to simplify group management in social networking sites, so as to help users set up their privacy policies. Group organization may help users set privacy settings on newly added content, or for new users joining social circles.

We envision several extensions of the current approach. First, an extensive user-centric study of the proposed techniques may be needed, to help further assess the practical value of the current solution, and guide the next steps of our research. Next, we would like to study how to select minimal features for privacy inference, rather than resort to common features. Identifying the features influential to privacy decisions would help optimize the algorithm, both with respect to accuracy and performance. Finally, we would like to study possible approaches to help users collectively control shared content belonging to group.

REFERENCES

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

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.

[3] E. A. Alessandra Mazzia, Kristen LeFevre, April 2011. UM Tech Report #CSE-TR-570-11.

[4] C. Blog. Exclusive: The next facebook privacy scandal., http://news.cnet.com/8301-13739_3-9854409-46.html, 2008.

[5] J. Bonneau, J. Anderson, and L. Church. Privacy suites: shared privacy for social networks. In *Symposium on Usable Privacy and Security*, 2009.

[6] J. Bonneau, J. Anderson, and G. Danezis. Prying data out of a social network. In *ASONAM: International Conference on Advances in Social Network Analysis and Mining*, pages 249–254, 2009.

[7] K. Burbary. Facebook demographics revisited – 2011 statistics. 2011.

[8] B. Carminati and E. Ferrari. Collaborative access control in on-line social networks. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on*, pages 231 –240, oct. 2011.

[9] W. Chronicle. Study raises new privacy concerns about facebook., http://chronicle.com/free/2008/02/1489n.htm, 2008.

[10] A. K. Fabeah Adu-Oppong, Casey Gardiner and P. Tsang. Social circles: Tackling privacy in social networks. In *Symposium On Usable Privacy and Security*, 2008.

[11] L. Fang and K. LeFevre. Privacy wizards for social networking sites. In *19th international conference on World Wide Web (WWW 2010)*, pages 351–360, New York, NY, USA, 2010. ACM.

[12] G. Hogben. Security issues and recommendations for online social networks. *ENISA Position Paper N.1*, 2007.

[13] H. Hu and G.-J. Ahn. Multiparty authorization framework for data sharing in online social networks. In *DBSec*, pages 29–43, 2011.

[14] C. E. Jacobs, A. Finkelstein, and D. Salesin. Fast multiresolution image querying. In *SIGGRAPH*, pages 277–286, 1995.

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

[16] K. Liu and E. Terzi. A framework for computing the privacy scores of users in online social networks. In *IEEE International Conference on Data Mining*, pages 288–297, 2009.

[17] M. Irvine. Social networking applications can pose security risks. *Associated Press*, April 2008.

[18] E. M. Maximilien, T. Grandison, T. Sun, D. Richardson, S. Guo, and K. Liu. Privacy-as-a-service: Models, algorithms, and results on the Facebook platform. Proceedings of Web 2.0 Security and Privacy (in conjunction with the IEEE Symposium on Security and Privacy)", 2009.

[19] D. Metzler, Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel. Similarity measures for tracking information flow. In *Proc. of the ACM international conference on Information and knowledge management (CIKM)*, pages 517–524, 2005.

[20] R. Ravichandran, M. Benisch, P. Kelley, and N. Sadeh. Capturing social networking privacy preferences. In *Symposium Of Usable Privacy and Security*, 2009.

[21] A. C. Squicciarini, S. Sundareswaran, D. Lin, and J. Wede. A3p: adaptive policy prediction for shared images over popular content sharing sites. In *HT*, pages 261–270, 2011.

[22] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 531–540, New York, NY, USA, 2009. ACM.