Ontology-based Privacy Setting Transfer Scheme on Social Networking Systems

Chen-Yu Lee[1], **Krishna M. Kavi**[1], **and Mahadevan Gomathisankaran**[1]
[1]Department of Computer Science and Engineering, University of North Texas, Denton, TX 76203, USA

Abstract—*In this age of eternal connectedness, using various social networks, privacy is an important problem that needs to be transparent. Even though social networks provide a user the ability to control their privacy settings, it is often difficult to get similar privacy settings across different social network systems. This is because: (a) privacy settings rely on complicated privacy rules which define the access control to different elements by different groups; (b) the terminology used varies in different social networking systems; (c) for a typical user, it is hard to set all the privacy settings as desired due to the complicated navigation through the social networking sites. Our goal is to design a framework that enables the transfer of privacy settings among social networking systems. We collect a user's privacy settings for many social services and store them in an ontology database. When a user registers on a new social network, our system provides recommendations of settings for it based on the user preferences as indicated by the settings in other services. Our framework covers personal privacy settings and the settings of the relationships of groups/tags and other elements.*

Keywords: social networking systems, privacy settings, ontologyocial networking systems, privacy settings, ontologys

1. Introduction

Social networking system and services have become an important part of on-line world for most people. Social networks allow users to share information among friends, groups, and companies instantaneously around the world. While sharing information is an important social phenomenon, the risk of losing privacy increases. Furthermore, the unsuspecting users may be prone to identity theft[1], password disclosure[2], account cracking[3], and so on.

Facebook[4] is a popular social networking service that has more than 1.19 billion users, as of September 2013, and is still growing [1]. For the past few years, Facebook allowed users to control their privacy settings in terms of sharing personal information, digital objects and other information, with other uses and third party services. In June 2012 Consumer Reports magazine reported that at least 13 million users had never set, or knew about Facebook's privacy tools and 28% of users shared all, or almost all of their wall posts with a wider public than their friends and sometimes to the entire public [2]. In general, we feel that users who have accounts on multiple social networks would like to have similar privacy settings. It is often difficult to get similar settings across different social network systems for three reasons: (a) privacy settings rely on complicated rules which control the access to different elements by different groups; (b) the terminology used varies greatly from one social network to another; (c) for a typical user, it is hard to set all the privacy settings as desired due to the complex navigation through the social networking sites. In this paper, we propose a framework that enables the transfer of privacy settings between social networking systems. We collect a user's privacy settings on many social services and store them in an ontology database. When the user registers on a new social service, our system provides recommendations for settings for the new one based on their setting on other services. Our framework covers personal privacy settings and the settings of the relationships of groups/tags and other elements.

The rest of the paper is organized as follows. Section 2 discusses research that is closely related to ours. Section 3 introduces our ontology model of security and privacy on social network systems and the privacy permission model is described in section 4. Section 5 presents the privacy transfer scheme and our experimental prototype is explained in section 6.

2. Related Works

Research on ontology-based privacy control started with development of ontologies for access control on social networking services. Kruk et al. proposed a Friend-of-a-Friend (FOAF) model which describes the social relationship as a directed graph [3]. FOAF-Realm is one of the earliest schemes to apply the FOAF ontology model for making decisions on resource access control according to the friendship levels. In Carminati's model [4], each authorization rule is designed subject to the type, depth, and trust level of the

[1] Social Thievery: Will Your Tweets Get You Robbed? Available: http://mashable.com/2011/11/01/social-theivery-infographic/

[2] Facebook ID Can Be Hack by Stealing Security Question - Answer, Available: http://hackw0rm.blogspot.com/2013/03/facebook-id-can-be-hack-by-stealing.html

[3] The hacker who broke into Mark Zuckerberg's Facebook page will get a $12,000 reward from online donors. Available: http://www.dailymail.co.uk

[4] https://www.facebook.com/

relationship which is represented in OWL. Villegas et al. [5] proposed a personal data access control (PDAC) scheme to classify the community of users into three parts: acceptance, attestation, and rejection using the ”trusted distance” measure. The trusted distance is measured by the relation hops between users and the other experiential information. Finin and Elahi [6] relied on role-based access control (RBAC) policies in social environment using OWL [7]. In 2009, Carminati’s framework defined access control policy, filtering policy, and admin policy encoded in Semantic Web Rule Language (SWRL) [8]. Masoumzadeh’s OSNAC system supports both user and system level authorization. It defined three main concepts: DigitalObject, Person, and Event in user level authorization [9]. Li et al. proposed a SPAC system, which extracts the privacy configuration patterns from each user’s profile and privacy settings using a semantics-enhanced K-Nearest Neighbors (K-NN) classification algorithm. The system predicts the privacy setting for new friends based on the patterns [10]. Shehab et al. presented an access control framework which enables users to specify shared data attributes and use the shared data as to manage third party applications in social networking services in 2012 [11].

In 2013, Masoumzadeh proposed a policy analysis framework to theoretically reason the missing pieces of policies and controls [12]. Kayes et al. proposed an ontology-based social ecosystem data model to generate platform-independent default privacy policy settings for each relationship group of a user according to multiple types of social interactions captured from various sources on a user’s devices [13]. Masoumzadeh and Kayes’ works inspired us to explore the development of a framework that allows transfer of privacy settings from one social network system to another.

It is not uncommon for people to have memberships in multiple social networks, and there are many services providing the ability to manage profile settings and to integrate the social medium on multiple social network sites such as Atomkeep[5]. However, the previous works are used to perform the access control in a social site, but they cannot solve the problem of porting privacy settings from one social site to another. A typical user does not want to learn how to manage settings in every new social network. Our framework aims to address this need.

3. Modeling Social Networking System Information

Our model of a generic social networking service consists of three parts: user, digital objects and provider. Our model is expressed in the Web Ontology Language (OWL2) in which each concept in the social service is modelled by an abstract object *class*. The relationships between classes are captured by *object properties*, and the relationship between classes and data values are captured by *data properties*.

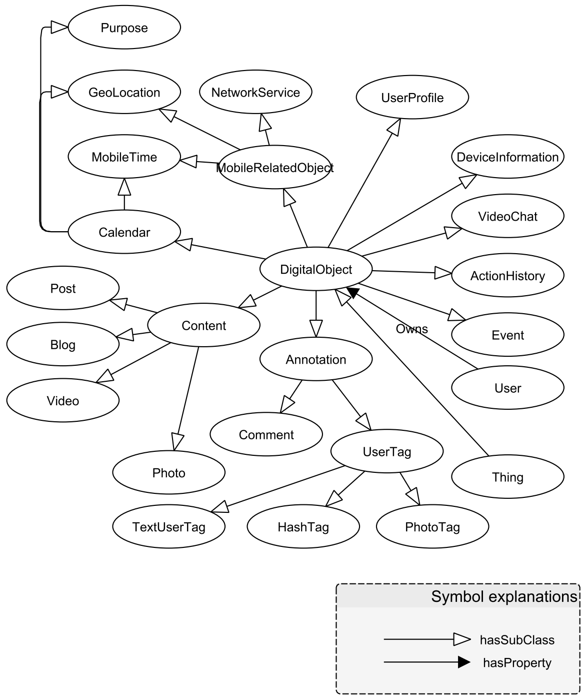


Fig. 1: The `DigitalObject` class in our ontology model

3.1 Digital Object Model

To model a social system, it is important to model its content because the content is the target of all the access controls. In our OWL model, we take a `DigitalObject` class to capture the users’ content in the social networking service as shown in Fig. 1. The content of a user can be divided mainly into two parts: user activities and user profiles. In general, people post their messages, photos, or videos on sites and leave some comments, or place tags on friends’ posts. We model the former post as `Content` which contains `Blog`, `photo`, `video`, and `post`; the latter is modeled as `Annotation` which has `Comment`, `TextUserTag`, `PhotoUserTag`, `HashTag`, and `URI` subclasses. People also like to arrange schedules of activities which captured as `Calendar` composed of `GeoLocation`, `MobileTime` and `Purpose`. In addition, some information may be saved by social sites such as footprints of life, and activity history, which are modeled as `MobileRelatedObject`, `ActionHistory` in our model.

3.2 User Profile Model

While exploring the issues of personal privacy, in addition to the content on the social networking site, a

[5] http://atomkeep.com/

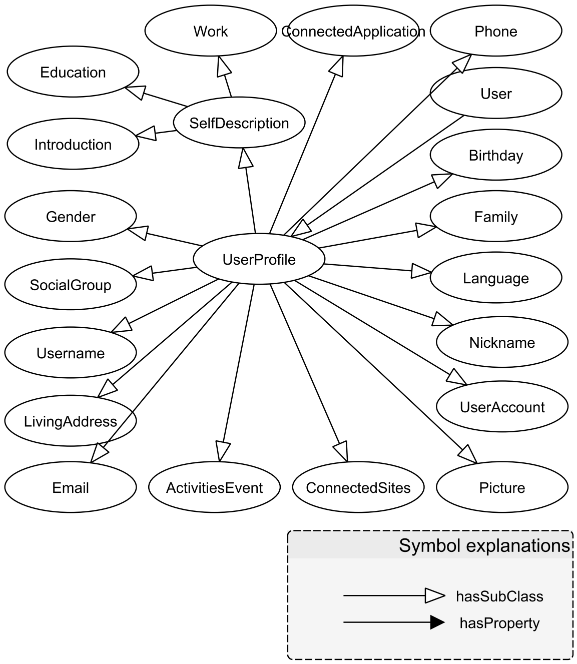


Fig. 2: The User class in our ontology model

user's personal information is critical. Personal information, or user profile, is a special content in the system because it is an important reference for people to know with whom they are interacting in social networks. Therefore, the publicity of the user profile should be controlled carefully. In general, the user profile contains many pieces of personal information such as photo, email, birthday, gender, phone, address, and other information related to the user's background like education, language, and work experiences. To capture all of a UserProfile, we model UserAccount, UserName, Picture, Email, Birthday, Nickname, Phone, Gender, Language and LivingAddress which is different from the concept of location. We also design some classes for personal experience like Work, Education, and SocialGroup. The user profile model is shown in Fig.2.

4. Modeling Privacy Sensitive Permissions

Personal privacy is guaranteed by access control policies which determine what information is revealed to whom. In our framework by default the data is deemed private to anyone except its owner. We design some access control properties to present the sharing status of each content. For example,

1 Alice_Email isShared Bob
2 Alice_Email.email:"Alice@abc.com"

where Alice_Email is an instance of Email address of Alice. The statement allows Bob to share Alice's email, Alice_Email. In the ontology, the statement does not reveal if Bob also shares other emails of Alice or if Alice permitted sharing of any other type information with Bob. Our goal is to capture these types of information. To do this, we define the following properties with access control.

- Searchable: Indicates that the object is granted searchable access in a service provider's search engine.
- isShared: Denotes an object shared with a user, i.e., a user is granted access to the shared information. As in the previous example, the statement shows that Alice's mail is shared with Bob.
- Shareable: Shareable property holds when the object is granted shared access to at least one user.

  1 Shareable: $\exists$ isShared.User
- isTagged: Tag-adding is a popular interaction between users in social sites. They like to add tags to posts, photos, videos, and any other content objects. "The isTagged property implies that the object permits tagging by other users. For example, a photo owned by Alice, which is tagged by Bob.

  1 Alice Owns Photo.df0075d2-b26e-4f6d-bbef-6df56ae8d653
  2 Bob Creates PhotoUserTag.5c934274-c53e-4f22-8bcb-2d8fa793a9ec
  3 Photo.df0075d2-b26e-4f6d-bbef-6df56ae8d653 isTagged PhotoUserTag.5c934274-c53e-4f22-8bcb-2d8fa793a9ec
  4 PhotoUserTag.5c934274-c53e-4f22-8bcb-2d8fa793a9ec.User: Bob

  where "Photo.df0075d2-b26e-4f6d-bbef-6df56ae8d653" is an instance of user photo, "PhotoUserTag.5c934274-c53e-4f22-8bcb-2d8fa793a9ec" is an instance of PhotoUserTag created by Bob, and

  1 PhotoUserTag, TextUserTag, HashTag $\subseteq$ UserTag
- Taggable: An object with the Taggable property implies that users may add tags to the object.

  1 Taggable: $\exists$ isTagged.UserTag
- isLinked: People often add photos on their web pages or blogs to make them rich. The property indicates that the content object in the system is linked by a URI which is a local or a foreign link. For example, Alice's photo is linked.

  1 Alice Owns Photo.df0075d2-b26e-4f6d-bbef-6df56ae8d653
  2 Alice Owns URI.809bb690-6656-4bbf-b28b-c3c7ce86be0e
  3 URI.809bb690-6656-4bbf-b28b-c3c7ce86be0e.uri: "www.csrl.edu/photo/alice"

4 Photo.df0075d2-b26e-4f6d-bbef-6df56ae8d653 isLinked

- Linkable: The property holds if the object is granted linking capability.

  1 Linkable: ∃ isLinked

- IsCommented: Making comments on a friend's post, photos, or videos is a common operation in social systems, allowing the ability to chat asynchronously, express feelings, thinking, and other actions as they would say to each other in face-to-face meetings. But people may also decide not to permit comments on their posts. The property defines if commenting access is granted to the object or not. For example, Alice allows her posts to be commented by the public.

  1 Alice Owns Post.6a299789-1c78-4b4d-8746-044b4234c225

  2 Post.6a299789-1c78-4b4d-8746-044b4234c225 isCommented.User PublicUser

  where "Post.6a299789-1c78-4b4d-8746-044b4234c225" is an instance of Post owned by Alice and

  1 PublicUser: ∀ User

- Commentable: The property holds if the object has granted commenting capability to other users.

  1 Commentable: ∃ isCommented.User

When a social network system is widely adopted, the account service would be designed to be a public identity service for signal sign-on to provide the federated identity for authentication mechanism like OpenID [6]. In addition to those properties used for the user's inner social system, some properties are designed to restrict access by other applications or sites.

- is3rdPartyAPPShareable: The social networking system will build a platform for third-party application developers who may provide useful services to the users in the social system. The property allows access of the object by third party applications running inside the social system. The following statement, for example, allows a social game, The Sims, access to Alice's email information.

  1 "Alice@abc.com" is3rdPartyAPPShareable TheSims

- is3rdPartySiteShareable: The property is similar to is3rdPartyAPPShareable, but the subject is third party sites which are built as standalone services rather than services built on the inner social platform.
- is3rdPartySiteLoginable: This property is also like the previous two, but it focuses on user login. The user can determine whether the account in the social system is used as a federated identity for authentication of accounts on other sites. In the case of consent, the user can login to the target site using the social network account in the system.

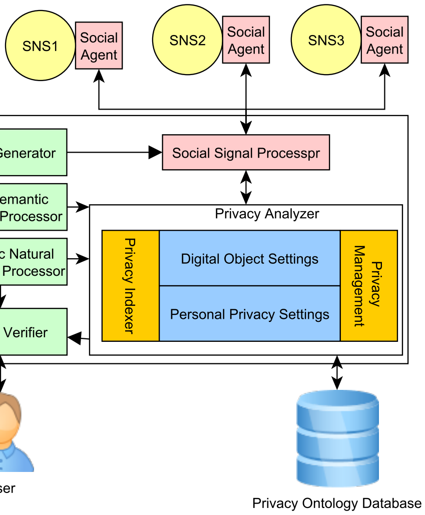


Fig. 3: The system architecture of our scheme.

5. Privacy Transfer Scheme

We propose a privacy transfer scheme which assists users in handling the privacy settings on social networking services with comparable preferences. The scheme is mainly divided into two parts: extraction and setting processes. The former extracts the privacy settings from one or more social networking services, and the latter sets the settings on a new service.

5.1 System Architecture

To realize our privacy transfer scheme, we designed several modules to extract information, analyse privacy settings, format the settings in the required manner, and recognize the similarity of terms used in different social networking services. Our system architecture is shown in Fig. 3 and each module is described below.

- Social agent: A third party application that extracts privacy information about a user and sends it back to our social signal processor. It works under the requirements: (1) the user registers the application and (2) allows access to privacy information. A user may share some information with friends and applications (including our social agent) or the user uses default settings (except for allowing our social agent). Our agent being a third application, it can gather the sharing information equal to that shared with friends or applications.
- Social signal processor: This module gathers privacy information sent from social agents executing on different

[6] http://openid.net/

social networking services and then sends them to the privacy analyzer. It is also responsible for placing the generated privacy setting on a new social networking service.

- Privacy analyser: The default setting of each index in the database is private, not shared with any object (friend or application). On the other hand, if privacy index is PUBLIC, the information is available to all objects. The privacy information is grouped into three parts: digital object, personal privacy, and access control settings. The analyser works by analysing the settings using our ontology model, setting the privacy indexes, and managing the user instances. Finally, the module saves user's instances in the privacy ontology database. For example, Alice shares her living address with Bob, the ontology database stores

  1 Alice `Owns` Alice_LivingAddress
  2 Alice_LivingAddress:1 Oak St. #1, Denton, Texas"
  3 Alice_LivingAddress `isShared` Bob

  where "1 Oak St. #1, Denton, Texas" is an instance of `LivingAddress`. Our system analyses the shared information, including personal privacy, to provide privacy setting recommendations.
- Setting generator: This module generates privacy setting scripts to be sent to the social signal processor for uploading them to a new social networking service.
- Word semantic similarity processor: As stated previously, the terms used by different social networking services are different, and sometimes they may use different terms to mean the same thing. We use word similarity between the terms and indexes [14].
- Semantic natural language processor: This module provides the ability to generate user friendly explanations for each generated privacy rule so that the user can approve selected settings.
- Privacy verifier: This module asks the user to verify the settings generated by setting generator. The settings can be uploaded to social networking services only if they are verified.

5.2 Extract Privacy Setting

Assume that the user already has a privacy setting on one social networking service, and he/she wants to transfer the setting to another one. The user has to register our application plug-in on the first service. Then the application begins the privacy setting extraction process following the workflow illustrated in Fig. 4.

Step 1 The **social agent**, installed on the social networking service as a third-party plug-in, extracts privacy information.

Step 2 **Social signal processor** gathers the social information from multiple social sites, if available, and sends them to privacy analyser for analysis.

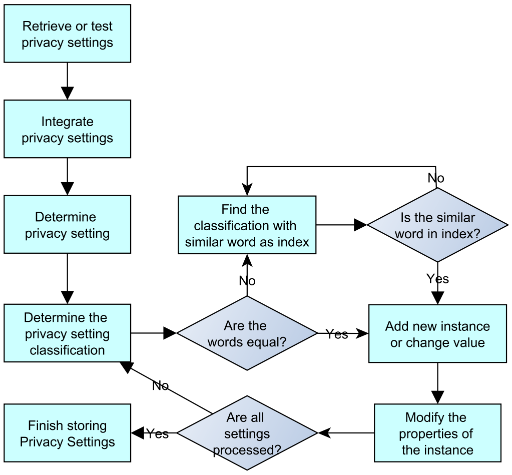

Retrieve or test privacy settings
Integrate privacy settings
Determine privacy setting
Find the classification with similar word as index
No
Is the similar word in index?
No
Yes
Determine the privacy setting classification
Are the words equal?
Yes
Add new instance or change value
No
Finish storing Privacy Settings
Yes
Are all settings processed?
Modify the properties of the instance

Fig. 4: The workflow of extracting the privacy setting from social networking services.

Step 3 **Privacy analyser** first distinguishes the privacy settings from received information and analyses the privacy setting according to the ontology indexes such as `Birthday`, `Gender`, etc. If it matches, the analyser adds a new instance of the classes or changes the value of the existing instances. If it does not match, the analyser finds a similar word as an index and then adds an instance to the index class. Usually, a pair of instances are added/modified for each privacy rule.

Step 4 Store the instances and properties in our ontology database. If all the privacy settings are not processed completely, go back to **Step 3**.

5.3 Set Privacy Setting

Suppose a user registers with a social networking service and installs our plugin application. Our scheme can then transfer his/her privacy settings to a new social networking service as described in the workflow shown in Fig. 6.

Step 1 When the social signal processor receives the request to transfer privacy settings onto the target social networking service, it asks the privacy analyzer to collect the settings from the database.

Step 2 The privacy analyzer collects the privacy information of the user for the target service and then sends the information to the privacy verifier for verification.

Step 3 The **privacy verifier** adds annotations for each privacy setting for the user to verify.

Step 4 The user can read the annotations to understand the corresponding settings and determine if they fit his/her preferences. For example, the annotations detail which privacy information is shared with which friend, group, or application, or if

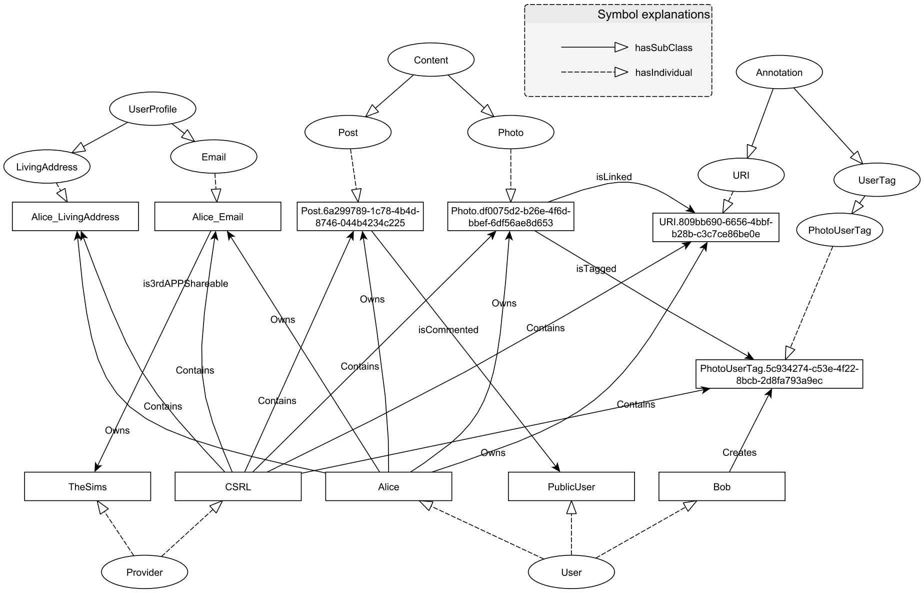


Fig. 5: A part of the privacy instances in our system which is described in section 4.

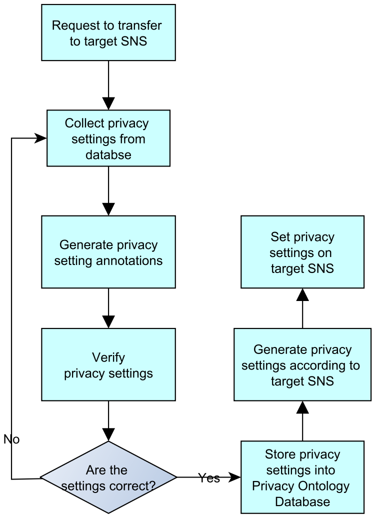

Request to transfer to target SNS
Collect privacy settings from databse
Generate privacy setting annotations
Verify privacy settings
Are the settings correct?
No
Yes
Store privacy settings into Privacy Ontology Database
Generate privacy settings according to target SNS
Set privacy settings on target SNS

Fig. 6: The workflow of setting the privacy setting on new social networking services.

the information is set public. If there are errors in the privacy settings, the user can correct them manually and go back to **Step 3**.

Step 5 The privacy analyser stores the correct privacy settings into the database.

Step 6 The privacy analyser sends the settings to the social signal processor to generate privacy settings for the target social networking service.

Step 7 The social signal processor sends the generated privacy script to the social agent which causes the equivalent privacy settings to be set on the target social networking service.

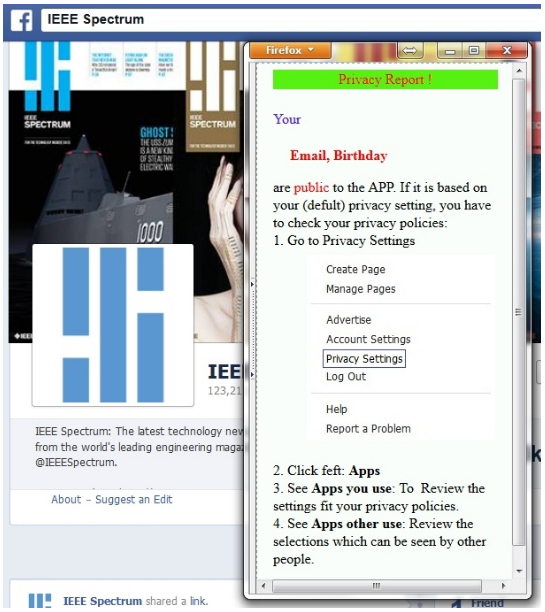
6. Implementation

6.1 Prototype

Our ontology-based scheme is implemented by the Portégé and Drupal platform to examine the performance of transferring a user's privacy settings. Drupal is an open source content management system that provides modules to build a social networking system. It also provides an application programming interface (API) for developers to create third party applications on the constructed social network system.

Fig. 5 shows a part of the privacy instances in our system which is described in section 4. Each ellipse is a class and each rectangle is an instance of the corresponding class. The arrow with a solid line indicates a subclass: e.g., `Content` has two subclasses `Post` and `Photo`. The arrow with a dashed line indicates individuals: e.g., `User` class has three individuals `Alice`, `Bob`, and `PublicUser`. The prototype stores the privacy settings of users on CSRL [7] as a service provider in our prototype. Because of a large number of individuals for some classes, especially `Content` and `Annotation`, they are named with a Universally Unique IDentifier (UUID) to prevent name conflicts: e.g., Post.6a299789-1c78-4b4d-8746-044b4234c225.

[7] CSRL stands for Computer Systems Research Laboratory.



Privacy Report !

Your

Email, Birthday

are public to the APP. If it is based on your (defult) privacy setting, you have to check your privacy policies:
1. Go to Privacy Settings

2. Click feft: **Apps**
3. See **Apps you use**: To Review the settings fit your privacy policies.
4. See **Apps other use**: Review the selections which can be seen by other people.

Fig. 7: An experiment for Facebook. The prototype is built on Facebook platform to notice users' privacy settings.

6.2 The Restriction on Public Social Networking Services

Our goal is to apply our scheme on real social networking services such as Facebook, Twitter, Google+, and so on, but at this time this is not feasible due to these restrictions:

- Third party application restriction: The third party application on public social networking services can only obtain limited information. Due to the privacy management and protection, only limited general personal information (which may include user name, friend list, location, time, etc.) is accessible to third party applications.
- Independent application restriction: The API of the service provider normally does not provide any information from one application to other applications, since each application is assumed to be independent on the platform. Thus it is difficult to understand which user information is shared with another application, and further to provide some privacy recommendations.

For these reasons, our scheme applied in a real social networking service, such as Facebook, can only extract the privacy information that is shared by users. However, this is adequate to demonstrate our framework. Our App on Facebook can access the user's email and birthday because the default privacy setting in Facebook is different from the setting stored in our system. In this case, our app produces a pop-up window as shown in Fig. 7. The user can learn how to change the setting by following our instructions step by step. We believe, therefore, users can then discovery the correct place to easily make the equivalent privacy setting.

7. Conclusion

Since Facebook became popular in social networking, there are more companies providing their own social networking services, including Google+, Qzone, Tumblr, and so on. It is unreasonable to expect users to acquaint themselves with every service's specific process for making various privacy settings.. We proposed a privacy transfer scheme to alleviate this problem. Our scheme not only provides recommendations to users on selecting their privacy settings, it also provides the ability for users to store and manage these settings. Users may have different privacy settings in different social networking services for different purposes, but if they desire to accept the similar settings, our scheme provides some recommendations and step-by-step guidance for them. In the future, we will extend our scheme to privacy management on mobile devices where large amounts of personal information are most commonly stored. Our goal is to develop a novel way of extracting and migrating privacy settings among public social networking services by overcoming existing, site-specific barriers to the process.

Acknowledgment

This research is supported in part by the NSF Net-centric and Cloud Software and Systems Industry/University Cooperative Research Center and NSF award 1128344.

References

[1] "Facebook reports third quarter 2013 results. facebook," 2013.
[2] "Facebook & your privacy: Who sees the data you share on the biggest social network?" *COMM. REP. MAG.*, 2012.
[3] S. R. Kruk, "Foaf-realm: control your friends' access to resources," in *Proc. FOAF Workshop*, 2004.
[4] B. Carminati, E. Ferrari, and A. Perego, "Rule-based access control for social networks," in *Proc. OTM'06*, 2006, pp. 1743–1744.
[5] W. Villegas, B. Ali, and M. Maheswaran, "An access control scheme for protecting personal data," in *Proc. PST'08*, 2008, pp. 24–35.
[6] A. J. T. Finin, L. Kagal, R. S. J. Niu, W. Winsborough, and B. Thuraisingham, "Rowlbac - representing role based access control in owl," in *Proc. SACMAT'08*, 2008, pp. 73–82.
[7] N. Elahi, M. M. R. Chowdhury, and J. Noll, "Semantic access control in web based communities," in *Proc. ICCGI'08*, 2008, pp. 131–136.
[8] B. Carminati, E. Ferrari, and A. Perego, "Enforcing access control in web-based social networks," *ACM T. INFORM. SYST. SE.*, vol. 13, no. 1, pp. 6:1–6:38, 2009.
[9] A. Masoumzadeh and J. Joshi, "Ontology-based access control for social network systems," *INT J. INF. PRIV. SECUR. INTEGRITY*, vol. 1, no. 1, pp. 59–78, 2011.
[10] Q. Li, J. Li, H. Wang, and A. Ginjala, "Semantics-enhanced privacy recommendation for social networking sites," in *Proc. TrustCom'11*, 2011, pp. 226–233.
[11] M. Shehab, A. Squicciarini, G.-J. Ahn, and I. Kokkinou, "Access control for online social networks third party applications," *COMPUT. SECUR.*, vol. 31, no. 8, pp. 897–911, 2012.
[12] A. Masoumzadeh and J. Joshi, "Privacy settings in social networking systems: What you cannot control," in *Proc. ASIA CCS'13*, 2013, pp. 149–154.
[13] I. Kayes and A. Iamnitchi, "Out of the wild: On generating default policies in social ecosystems," in *Proc. IEEE ICC'13*, 2013.
[14] L. Han, T. Finin, P. McNamee, A. Joshi, and Y. Yesha, "Improving word similarity by augmenting pmi with estimates of word polysemy," *IEEE T. KNOWL. DATA. EN.*, vol. 25, no. 6, pp. 1307–1322, 2013.