Privacy Settings in Social Networking Systems: What You Cannot Control

Amirreza Masoumzadeh
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, USA
amirreza@sis.pitt.edu

James Joshi
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, USA
jjoshi@pitt.edu

ABSTRACT

In this paper, we propose a framework to formally analyze what privacy-sensitive information is protected by the stated policies of a Social Networking System (SNS), based on an expression of ideal protection policies for a user. Our ontology-based framework can capture complex and fine-grained privacy-sensitive information in SNSs, and find out missing policies, given a user's ideal policies, and SNS's privacy settings and described system policies. We propose notions of policy completeness for SNSs to facilitate such an analysis. Our case study of using this approach on Facebook shows that we can effectively identify important missing policies.

Categories and Subject Descriptors

K.4.1 [**COMPUTERS AND SOCIETY**]: Public Policy Issues—*Privacy*

Keywords

Social Networking Systems, Privacy Settings, Privacy Control Policies

1. INTRODUCTION

In a typical Social Networking System (SNS) such as Facebook, there exist several privacy settings that can be configured by a user in order to control others' access to the information related to her. These settings are in fact equivalent to access control policies that are expressed by the users for the respective digital objects. Such privacy settings are by no means complete in the sense that they do not control access to all the potentially privacy-sensitive information about a user. That is the case even for Facebook, which has fairly the most extensive set of privacy settings among SNSs. Access to the rest of the information related to the user is governed by a set of fixed rules set by the SNS itself. We call these system defined policies and the user-configurable privacy settings together as *privacy control policies* in an SNS.

ASIA CCS'13, May 8–10, 2013, Hangzhou, China.

There is a major issue with the current practice of privacy control policies in SNSs such as Facebook. The current privacy settings do not provide users with adequate power to protect their privacy-sensitive information. Moreover, the system defined policies enforced by an SNS are not clearly described to the users. Therefore, users are unsure about what to expect from the system. Users need to learn about them either by harvesting help pages in the SNS or by observing the system's behavior. Worse is that, since the system defined policies are not well documented, SNSs can modify them without users noticing it and put them at great risk of privacy violations.

In order to identify the privacy risks users are dealing with in such an abovementioned ecosystem, we propose an approach to formally reason about *completeness* of privacy control policies in SNSs. Our notion of completeness ensures that privacy control policies are applicable to every piece of information related to a user. Therefore, the user can clearly expect to know that how her information is protected. Most of the recent literature on policy analysis focuses on XACML policies [5, 6], which are not suitable for representing complex policies in SNSs such as "*Who can see posts you've been tagged in on your timeline?*". A separate body of literature has focused on modeling access control policies for SNSs. Most of those models formalize specification of access subjects based on the type and distance of their relationships to an object's owner [2, 3]. However, such models cannot consider protection of complex privacy-sensitive resources in SNSs as required for our analysis purpose. A few recent works approach the access control modeling problem in SNSs by using Semantic Web technologies [1, 7, 8], which shows great promise in capturing the complex and fine-grained access control policy requirements in these systems. Like in [7, 8] we approach this issue by modeling the information contained in an SNS using an ontology. However, we employ a more rigorous approach with regards to modeling data constraints in SNSs, as they are vital to provide sound analysis of privacy control policies.

In this work, we propose a framework to formally reason about the completeness of privacy control policies in an SNS. To the best of our knowledge, we provide the first of its kind policy analysis approach for SNSs that can theoretically reason about the missing pieces of policies and controls in SNSs. Such a systematic approach can tell the SNS users if their expectations about privacy control (or system defined policies in place) is provided by an SNS, and ultimately help the developers of such systems to resolve these issues. In summary, we make the following contributions:

- We propose a fine-grained ontology-based approach to model information in SNSs, including data integrity constraints, that are necessary for the reasoning tasks in our framework.
- We present an approach to specify permissions in such an ontology-based knowledge base, using which we reason about completeness of privacy control policies provided by an SNS, and detect missing policies.
- Using our framework, we analyze Facebook's privacy control policies as a case-study, and present our findings with regards to its completeness.

The rest of the paper is organized as follows. We propose a fine-grained model of information in SNSs, and an approach to specify privacy-sensitive permissions in these systems in Sections 2 and 3, respectively. In Section 4, we formally define our notion of completeness, and show how our framework can analyze policies in an SNS based on that. In Section 5, we demonstrate the applicability of our approach by providing a case study, and conclude the paper in Section 6.

2. MODELING SNS INFORMATION

In this section, we propose a model of SNS information using OWL [4] as our modeling language. We first discuss a generic information model for SNSs and then explain in more detail an ontology for Facebook. Clearly, a similar ontology can be developed for other SNSs.

2.1 Basic Concepts

We model SNS information as a set of users, digital objects, and data values, which are related to each other by relationships. In OWL terminology, we model the first two concepts using *classes* as abstract objects. The relationships between objects are captured using the following *object properties*: between users to define the social network (e.g., friendship, following, etc.), between digital objects (e.g., a comment that is related to a photo), or between users and digital objects (e.g., a user may own a photo). Moreover, class objects can be related to data values using *data properties*, e.g., the relationship between a photo and its binary content. We model annotations as a special class of digital objects. Annotations are objects that annotates one object with another object. Comments and photo tags are two common ways of annotation. For instance, a photo tag can annotate a photo with a user. As another example, a check-in in Foursquare is an annotation that annotates a venue (place) with a user, which may also be associated with a time stamp. These are the minimum concepts and relationship types that we capture for an SNS; they can be easily extended depending on the needs of an SNS.

Figure 1 depicts our proposed ontology for Facebook. Note that many details, especially restrictions, are not captured in the figure or in the following description. Entity is the root class to our ontology (a subclass of owl:Thing, which is OWL's built-in most general class). It is specialized by User and DigitalObject. There is a predefined individual in class User, i.e., me, for whom policy analysis is performed. The isFriendOf object property expresses friendship relationship between instances of User. Data properties such as hasFullname associate data values to a User. The owns object property defines a User as the owner of a DigitalObject. DigitalObject is the union of four major subclasses: Content, Wall, Event, and Annotation. Content represents an object that has data content such as a photo or a text (specified using hasContent data property). Classes Wall and Event correspond to profile wall page and events that users can attend in Facebook. Annotation, as mentioned before, represents objects that instead of directly carrying content, annotate a DigitalObject (e.g., a wall or a photo) with an Entity (e.g., a wallpost or a comment), using object properties annotates and annotatesWith, respectively. Actual forms of annotations are defined as its subclasses: Comment, UserTag, and WallPost. A Comment annotates a Commentable (a class which is the union of Photo and WallPost) with a Text. A UserTag annotates a UserTaggable (a class which is the union of Photo and Text) with a User.

2.2 Restrictions

It is crucial to accurately model the data constraints in an SNS knowledge base, in order to facilitate meaningful policy analysis. Any inaccuracy will result in false positives/negatives in our analysis. We employ the following restriction features in OWL to model such data constraints:

Disjoint Union of Subclasses: We ensure our class hierarchy is captured completely by defining each class as disjoint union of its subclasses, i.e., union of subclasses which are pair-wise disjoint. For instance, Annotation is equivalent to the disjoint union of Comment, UserTag, and WallPost.

Property Domain/Range: The domain and range of a property can be restricted to specific classes, e.g., owns has User and DigitalObject as its domain and range, respectively.

Property Characteristics: OWL supports several restrictions to accurately model property constraints: functional, inverse functional (i.e., the inverse of the relation is a function), transitive, symmetric, asymmetric, reflexive, and irreflexive. For instance, isFriendOf is defined as irreflexive; hence, no user can be a friend of herself.

Class Property Restriction: OWL supports existential, universal, cardinality, and value constraints for properties when applied to a class. For instance, WallPost is defined equivalent to a class that annotates exactly one Wall.

3. MODELING PRIVACY-SENSITIVE PERMISSIONS

For our analysis purpose, we do not aim at analyzing the expressive power of the SNS privacy settings in terms of characterizing the access subjects. Instead, we focus on capturing the fine-granularity of the protected objects in an SNS, as described in this section.

3.1 Properties as Protected Resources

The ontology modeling approach proposed in Section 2 captures SNS information in the forms of classes and properties. We argue that an individual of a class does not represent any privacy-sensitive information unless its properties are considered. In other words, the essential knowledge is captured by the triples that represent properties between two individuals (object properties), or between an individual and a data value (data properties). Based on this observation, we consider such triples as privacy-sensitive information that needs to be protected.

We elaborate on this further using the example ontology in Figure 2. Assume Alice has a photo, in which Bob has been tagged by Carol: Alice owns Photo1, and Carol owns PhotoUserTag1 which annotates Photo1 with Bob. By con-

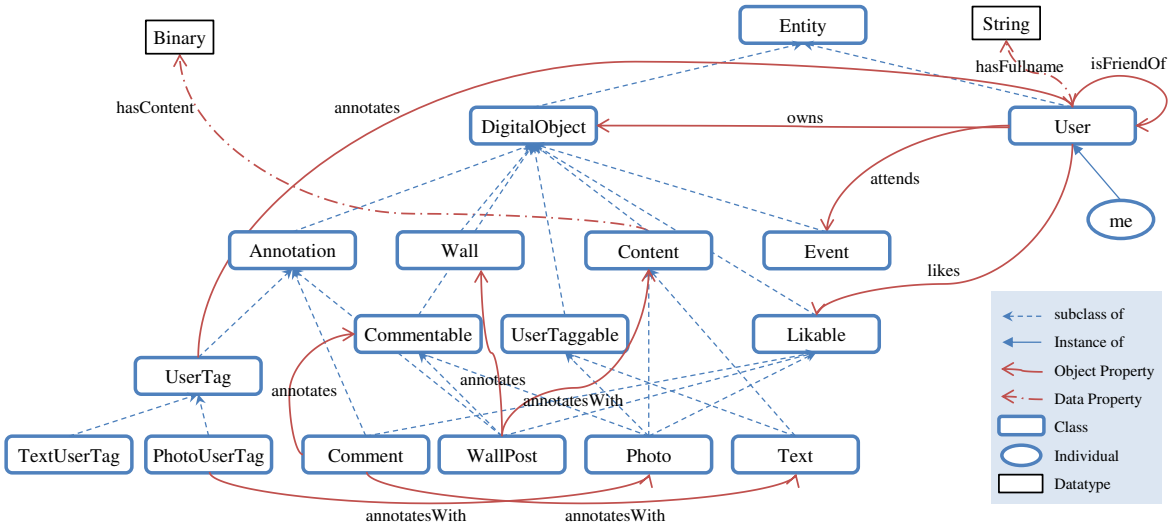

Figure 1: A Model of Concepts and Properties for Facebook

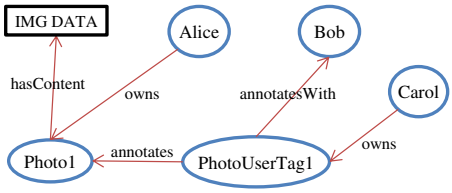

Figure 2: A PhotoUserTag Example

sidering the object and data properties in this example as protected resources, we can control the access to privacy-sensitive information. We assume a simple and effective policy authority scheme: the owners of the endpoints of each property are eligible to define policy for that property. Therefore, for instance, visibility of the tag can be controlled using relationships PhotoUserTag1 annotates Photo1 (by Alice or Carol) and PhotoUserTag1 annotatesWith Bob (by Bob or Carol). Carol can control revealing the fact that she has created this tag by controlling Carol owns PhotoUserTag1. Finally, Alice can control the association of the actual binary content of the photo using relationship Photo1 hasContent IMG-DATA. Various permissions can be specified with respect to the three basic possible actions on these resources: selection, insertion, and deletion. For example, the insertion of a tag can be controlled in a similar fashion.

3.2 Reification of Properties

In order to specify permissions, we need to characterize classes of relationships in an ontology based on certain restrictions. However, OWL does not support such expressions about relationships. We use the concept of reification to overcome this limitation. We reify each object/data property in our SNS ontology as a class in OWL. Figure 3 depicts the policy ontology corresponding to the ontology presented in Section 2. Class ReifiedProperty is the root to all such classes, with the two subclasses ReifiedObjectProperty and ReifiedDataProperty, which are the disjoint union of the corresponding reified property classes. We define two special object properties, ropSbj and ropObj, that relate a reified object property to its subject and object, respectively. Analogously, the special object and data properties rdpSbj and rdpData relate a reified data property to its subject and data value, respectively.

The reified property classes will essentially replace SNS ontology properties that were described in Section 2. Therefore, they must conform to the information domain constraints that we considered in designing the ontology. We demonstrate with an example how a property restriction presented in 2.2 can be translated for reified properties. Let p be an object property in the SNS ontology, and RPp be its corresponding reified class. If p is a functional property, such a characteristic can be ensured for RPp by the following restriction: owl:Thing $\sqsubseteq$ inverse ropSbj max 1 RPp.

3.3 Representing Permissions

A permission can be represented by a set of protected resources and corresponding action(s). The set of possible actions on a resource include viewing, creating, or removing it from an SNS knowledge base. We define class Action which consists of three individual members select, insert, and delete, respectively. We represent permissions by characterizing a subset of class Permission. The class Permission is related to class RefieidProperty using object property pResource to specify the resources. It is also related to class Action using object property pAction in order to specify the considered action(s) in the permission. As an example, consider the privacy setting in Facebook that states "*who can post to your timeline?*". The corresponding permission can be represented using the following specification:

S2 ≡(pAction value insert)
and (pResource some (RPannotates
and (ropObj some (Wall
and (inverse ropObj some (RPowns
and (ropSbj value me)))))))

In the above conjunctive expression, the first clause specifies the action as insertion. The second complex clause indicates that the resource should be of a reified property class RPannotate, where its object is a Wall, and that wall is

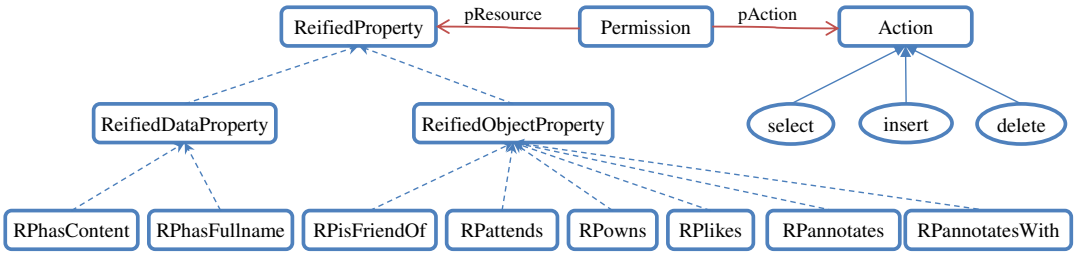

Figure 3: Representing Permission Specification in Ontology

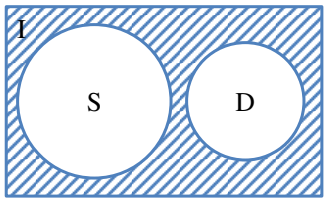

Figure 4: Permission Categories Space for an SNS

owned by me. In other words, the resource is the annotates relationship that annotates the Wall belonging to me.

4. ANALYZING PRIVACY CONTROL PERMISSIONS

We identify three categories of permissions in an SNS:

Privacy Setting Permissions These are captured by user-configurable privacy settings, which usually have dedicated control elements in the system.

Described System Permissions These consists of the permissions that are not configurable by users and their corresponding system-defined policy is well documented.

Ideal Permissions These refer to the permissions that a user ideally considers as privacy-sensitive, which may depend on the information model of the SNS.

Let S, D, and I, represent the classes of privacy setting, described system, and ideal permissions in an SNS, respectively. Figure 4 depicts these permission classes in relation to each other. The privacy setting permissions and described system permissions are disjoint by definition. The ideal permissions (the rectangle) is likely to cover both these classes. However, we note that there may exist parts of ideal permissions that are covered by neither privacy settings nor described system permissions. In other words, there may exist some privacy-sensitive permissions which are not mediated by a transparent policy. Our framework can reason about existence and nature of such permissions.

Our permission specification, as proposed in Section 3.3, is expressed in terms of classes in the ontology. Hence, we can leverage ontological reasoning power to compare and analyze such permissions. Let P and Q be two permission classes. An OWL DL reasoner can efficiently evaluate the following permission compositions:

- not P: denotes what P does not cover.
- P and Q: denotes what both P and Q cover.
- P or Q: denotes what either P or Q (or both) cover.

Using the above compositional semantics and subsumption re we can analyze permissions in an SNS. The class M $\equiv$ I and not (S or D) represents the permissions that users are missing from the picture. We further elaborate on such analysis in our case study in Section 5.

We propose two completeness privacy properties for an SNS based on this analysis. The following property ensures that users' ideal permissions are part of the privacy settings and can be controlled by them.

Definition 1 (Completely Controllable). *An SNS with its privacy setting permissions* S *is* completely controllable *with regards to ideal permissions* I*, if and only if* $\mathsf{I} \sqsubseteq \mathsf{S}$.

In practice, with a reasonable assumption about ideal permissions, the above notion of completeness cannot be satisfied in SNSs. The users may be overwhelmed by the complexity of the options, and the system may choose not to provide such controls due to various design considerations. The next notion of completeness is more practical.

Definition 2 (Completely Known). *An SNS with its privacy setting and described system permissions* $\langle \mathsf{S}, \mathsf{D} \rangle$*, is* completely known *with regards to ideal permissions* I*, if and only if* $\mathsf{I} \sqsubseteq (\mathsf{S} \text{ or } \mathsf{D})$.

The above notion of completeness verifies if the collection of privacy settings and described system permissions by an SNS covers the ideal permissions. If not, we can employ subsumption reasoning to find permissions that the user misses.

5. CASE STUDY: FACEBOOK PRIVACY CONTROL PERMISSIONS

In order to evaluate our framework, we analyze the privacy control policies in Facebook. We leverage the ontology described in Section 2, in which we refrain from completely modeling every detail in Facebook for brevity reasons. We believe the current ontology is representative enough to describe our methodology, and other features can be captured in a similar fashion.

Facebook provides a centralized dashboard for controlling privacy settings such as the visibility of the statuses, tagging, etc. Moreover, a user can determine the visibility of her profile information such as education, contact info, etc. We collected those settings and formulated their corresponding permissions. In Table 1, we list those settings either worded as exactly as in the settings' page or representative enough, and corresponding permissions. Note that the listed privacy settings are limited to the information that can be captured by our proposed ontology in Section 2. For instance, the setting "*Who can send you Facebook messages?*" is not listed since we did not include messages in our ontology. Our current ontology also does not model apps, or

	Privacy Setting	Corresponding Permission
S1	Profile attributes privacy (for each item)	(pAction value select) and (pResource some (rdpSbj value me))
S2	Who can post on your timeline?	(pAction value insert) and (pResource some (RPannotates and (ropObj some (Wall and (inverse ropObj some (RPowns and (ropSbj value me)))))))
S3	Who can see what others post on your timeline?	(pAction value select) and (pResource some (RPannotates and (ropObj some (Wall and (inverse ropObj some (RPowns and (ropSbj value me))))) and (ropSbj some (WallPost and (inverse ropObj some (RPowns and (ropSbj value me)))))))
S4	Who can see what you post? (per item)	(pAction value select) and (pResource some (RPhasContent and (rdpSbj some (inverse ropObj some (RPowns and (ropSbj value me))))))
S5	Review tags friends add to your own posts	(pAction value insert) and (pResource some (RPannotates and (ropObj some (inverse ropObj some (RPowns and (ropSbj value me)))) and (ropSbj some PhotoUserTag)))
S6	Visibility of photos (managed per album)	(pAction value select) and (pResource some (RPhasContent and (rdpSbj some (inverse ropObj some (RPowns and (ropSbj value me))))) or (RPowns and (ropSbj value me) and (ropObj some Photo)))

Table 1: Privacy Setting Permissions in Facebook

	Described System Policy	Corresponding Permission
D1	Who can see tags I make?	(pAction value select) and (pResource some (RPannotatesWith and (ropSbj some (UserTag and (inverse ropObj some (RPowns and (ropSbj value me)))))))
D2	Who can see that I'm tagged in a post?	(pAction value select) and (pResource some (RPannotatesWith and (ropObj value me)))
D3	Who can see a tag that someone added to my post?	(pAction value select) and (pResource some (RPannotates and (ropObj some (UserTaggable and (inverse ropObj some (RPowns and (ropSbj value me)))))))

Table 2: Described System Permissions in Facebook

the posts that are automatically made by the system on one's timeline when she is tagged in another post. Moreover, in order to avoid enumerating a tedious list of settings, we provide one privacy setting that accounts for all profile attributes (S1). That setting can cover permissions corresponding to a setting like "*Who can look you up using the email address or phone number you provided?*" as well as many other single visibility controls in front of every profile item under a user's *about page*.

Unfortunately, Facebook is not very transparent about its system defined policies. Some of the system defined policies are documented in the help pages, in the format of questions and answers, which is not easily accessible, unless one goes through all the pages. For instance, the following question is provided in the help page for tagging (and not under privacy!): "*When I tag someone in a photo, post or app activity, who can see it?*". The answer provided to this question explains that the audience selected for the post, the user tagged, etc., can see it. Table 2 lists some of the described system policies that we were able to harvest from Facebook's help pages, and that were related to the information captured in our ontology.

Based on our model of SNS information, our intuition is that a user should be able to control relationships that are about her. In terms of our proposed Facebook ontology, those includes properties that directly relate to the user, and the properties that relate to some digital objects owned by the user. We consider these as our ideal permissions for a user in an SNS, as shown in Table 3. Note that this is a fairly conservative ideal model in the sense that every related information to a user is considered as potentially privacy-sensitive. However, we see it as a safe and reasonable assumption.

Based on the abovementioned permissions, it is clear that Facebook does not satisfy *completely controllable* property (Definition 1) with regards to ideal policies listed in Table 3. The permissions in Table 2 are clearly not covered by the privacy settings, and we were able to verify this using our framework. For instance, $\mathsf{D1} \sqsubseteq \mathsf{I}$ and not S is satisfiable.

Our framework also shows negative result for Facebook with regards to the *completely known* property (Definition 2), i.e., $\mathsf{I} \sqsubseteq (\mathsf{S}$ or $\mathsf{D})$ is not satisfiable. In order to identify some of the missing policies, we formulated several permission classes (subclasses of ideal permissions), and tested against formula $\mathsf{M} \equiv \mathsf{I}$ and not (S or D), as explained in Section 4. We report in Table 4 some permissions that are missing in current Facebook privacy control policies according to our analysis. The table lists only a sample of permissions with selection as action. As expected, many missing permissions can be found for insertion and deletion permissions, that we did not include.

	Ideal Privacy Setting	Corresponding Permission
I1	Control whatever that relates to you	(pAction some Action) and (pResource some (ReifiedProperty and ((rdpSbj value me) or (ropObj value me) or (ropSbj valueuj me))))
I2	Control whatever that relates to something belonging to you	(pAction some Action) and (pResource some (ReifiedProperty and ((rdpSbj some (inverse ropObj some (RPowns and (ropSbj value me)))) or (ropObj some (inverse ropObj some (RPowns and (ropSbj value me)))) or (ropSbj some (inverse ropObj some (RPowns and (ropSbj value me)))))))

Table 3: Ideal Permissions for Facebook

	Missing Policy	Corresponding Permission
M1	Who can see that I have tagged someone?	(pAction value select) and (pResource some (RPowns and (ropObj some PhotoUserTag) and (ropSbj value me)))
M2	Who can see that I have liked something?	(pAction value select) and (pResource some (RPlikes and (ropSbj value me)))
M3	Who can see my comment on someone else's post?	(pAction value select) and (pResource some (RPannotates and (ropSbj some (Comment and (inverse ropObj some (RPowns and (ropSbj value me))))) and (ropObj some (inverse ropObj some (RPowns and (ropSbj some Others)))))
M4	Who can see if I am friend with someone?	(pAction value select) and (pResource some (RPisFriendOf and (ropSbj value me)))

Table 4: Sample Missing Permissions in Facebook

6. CONCLUSIONS

The privacy settings in existing SNSs such as Facebook do not provide users with adequate power to control their privacy-sensitive information. Furthermore, SNS system-defined policies are not typically clearly described and hence may not be easily understood by the users. To enable assessment and analysis of privacy protection achieved within an SNS environment, we proposed an analysis framework based on the ontology-based formulation of permissions in privacy control policies. We formulated completeness properties based on a given ideal specification of permissions and policies by an SNS in terms of privacy settings and described system policies. Our framework is capable of formally reasoning about such completeness properties and identifying undocumented and unclear practices of an SNS with regards to protecting privacy-sensitive information. We demonstrated the analysis power of our framework by performing a case study on analyzing Facebook's privacy control policies.

As future work, we plan to investigate efficient and effective algorithms to automate the process of identifying the missing policies. Moreover, we plan to extend our approach towards a comprehensive framework that can consider other components in the privacy control policies (e.g., subjects), and perform variety of privacy analysis tasks to support users and system developers.

Acknowledgments

This research has been partly supported by the U.S. National Science Foundation awards IIS-0545912 and DUE-0621274. We would like to thank Professor Elisa Bertino and our anonymous reviewers for their helpful comments.

7. REFERENCES

[1] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. A semantic web based framework for social network access control. In *Proc. 14th ACM Symposium on Access Control Models and Technologies*, pages 177–186. ACM, 2009.

[2] B. Carminati, E. Ferrari, and A. Perego. Enforcing access control in Web-based social networks. *ACM Trans. Inf. Syst. Secur.*, 13(1):1–38, Nov. 2009.

[3] P. W. Fong. Relationship-based access control: protection model and policy language. CODASPY '11, pages 191–202, San Antonio, TX, USA, 2011. ACM.

[4] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph. OWL 2 Web Ontology Language - Primer. `http://www.w3.org/TR/owl2-primer/`, 2009.

[5] V. Kolovski, J. Hendler, and B. Parsia. Analyzing web access control policies. In *Proc. 16th Int'l Conference on World Wide Web*, pages 677–686, 2007.

[6] D. Lin, P. Rao, E. Bertino, N. Li, and J. Lobo. EXAM: a comprehensive environment for the analysis of access control policies. *International Journal of Information Security*, 9(4):253–273, Aug. 2010.

[7] A. Masoumzadeh and J. Joshi. OSNAC: An Ontology-based Access Control Model for Social Networking Systems. In *Proc. 2nd IEEE Int'l Conference on Information Privacy, Security, Risk and Trust (PASSAT 2010)*, pages 751–759, 2010.

[8] A. Masoumzadeh and J. Joshi. Ontology-based access control for social network systems. *International Journal of Information Privacy, Security and Integrity (Special Issue: Selected Papers from PASSAT 2010)*, 1(1):59–78, Jan. 2011.