Providing Access Control to Online Photo Albums Based on Tags and Linked Data *

Ching-man Au Yeung[1], **Lalana Kagal**[2], **Nicholas Gibbins**[1], **Nigel Shadbolt**[1]
[1]School of Electronics and Computer Science, University of Southampton
Southampton, SO17 1BJ, United Kingdom
[2]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology
Cambridge, MA 02139, USA
cmay06r@ecs.soton.ac.uk, lkagal@csail.mit.edu, nmg@ecs.soton.ac.uk, nrs@ecs.soton.ac.uk

Abstract

While photo sharing sites such as Flickr provide efficient tools for setting up an online album, users who want to maintain a certain level of privacy are usually only provided with rudimentary access control. Given that descriptive tags are extensively used on photos, and that the Semantic Web provides a common means of sharing social network information as linked data, we believe better access control mechanism can be provided by combining the two. Based on this idea, we propose and describe in this paper a system which allows users to create expressive access control policies for their photos on the Web by using both tags and linked data.

Introduction

Sharing photos on the Web has become very popular among Web users nowadays. Web sites such as Flickr (http://www.flickr.com/) allow users to upload their photos and describe them using tags, which are descriptive terms chosen by the users as they like. While these Web sites promote sharing one's photos with other users on the Web, some users are also concerned about their privacy and may only want to share their photos with a certain group of people, instead with all other users on Web who are unknown to them (Miller and Edwards 2007).

Currently, mostly photo sharing sites only allow users to specify whether a photo is public, private or visible to their family members or friends. Users can only apply this setting to an individual photo or a particular set of photos. It is not possible to share photos with only, for example, one's colleagues or people who participated in a particular event. Users may also have to compile their lists of friends again if they move to another photo sharing site.

*The authors would like to thank Tim Berners-Lee and other members of the Decentralized Information Group (DIG) at CSAIL, MIT for contributing to the ideas of this project. The project was undertaken by the first author while he was a visiting student at DIG as part of an exchange programme supported by the Web Science Research Initiative (EPSRC Grant No. EP/F013604/1). This work was also supported in part by funding provided by the National Science Foundation (NSF Grant No. 6898398).

Given that tags are extensively used on photos on these Web sites, and that they provide rich information of what the photos are about, it is possible to provide better access control mechanism based on the tags assigned to the photos. In addition, Semantic Web linked data (Bizer et al. 2008) such as user profiles based on the FOAF (Friend-Of-A-Friend) ontology (Brickley and Miller 2007) can be used to specify one's social network on the Web which other applications can easily refer to. By combining tags and linked data, we believe a more flexible and comprehensive access control mechanism can be provided. For example, a user should be able to specify that his photos with the tag *wedding* and *party* can only be viewed by his friends which are mentioned in his FOAF profile.

We describe a system which combines tagging and linked data to provide access control to a person's photo collection on the Web. This system relies on the OpenID protocol (Recordon and Fitzpatrick 2006) for authentication, extends the Tag Ontology (Newman 2005) to represent tagging activities in a photo album, uses the AIR Policy Language (Kagal, Hanson, and Weitzner 2008) to specify access control policies, and uses the Tabulator (Berners-Lee et al. 2006) as the basis of the user interface for browsing photos (and their metadata in RDF) and specifying access control policies.

Background

This section provides some background information of our system, including linked data on the Semantic Web, ontologies for describe tagging activities on the Web, OpenID, the AIR Policy Language and the Tabulator.

Semantic Web and Linked Data

The Semantic Web is designed to provide a framework for describing data on the Web with machine-readable metadata. It also promotes linked data, the idea of connecting data on the Web by using their dereferenceable URIs. Recently, the use of RDF in publishing data of social interactions on the Web has become increasingly popular. For example, the FOAF ontology has been widely used by Web users to describe their social networks. Many organisations, such as the research groups which the authors of this paper are affiliated to, also publish information about their members in RDF. Other ontologies such as SIOC (Semantically-Interlinked Online Communities) (Breslin et al. 2005) and

SCOT (Semantic Cloud of Tags) (Kim et al. 2007) have also been proposed to be used for describing activities of online communities such as online forums and collaborative tagging systems. As the different roles of a user becomes more clearly defined in the Semantic Web, access control schemes which uses linked data will find it easier to determine whether the user is authorised to perform a particular task.

Ontologies of Tagging

Tagging is the act of assigning descriptive keywords to online resources such as bookmarks, photos and videos. It has been made popular by Web sites such as Delicious (http://delicious.com/) and Flickr as a means of organising and sharing resources on the Web. Several ontologies have been proposed to provide a formal conceptualisation of tagging and allow reuse of tagging data across different tagging systems. For example, Gruber (Gruber 2007) proposes that tagging is a five-place relation: `Tagging(object, tag, tagger, source, +/-)`. Newman (Newman 2005) proposes the Tag Ontology which uses the class of `Tagging` to bind together the user, the resource being tagged, the set of tags used and the time at which the tags are assigned. The SCOT project introduces the class of `TagCloud` to group together a set of tagging activities. A more comprehensive review of different ontologies of tagging can be found in (Kim et al. 2008).

The OpenID Authentication Protocol

OpenID is first proposed by Fitzpatrick (Recordon and Fitzpatrick 2006) in 2005. It is an authentication protocol which allows users to maintain a single digital identity that can be used to log on to different Web sites. To use OpenID, a user chooses a trusted OpenID provider with which he maintains a unique ID in the form of an URL. Other systems which want to authenticate the user will rely on this OpenID provider to perform authentication. The OpenID protocol is decentralised as any Web site can use OpenID to verify a user's identity without maintaining a centralised database. The FOAF ontology provides the property `foaf:openid` for specifying the OpenID of a `foaf:Person`. A user can then be verified if he is the person mentioned in a FOAF profile, thus provides a decentralised authentication method for Semantic Web applications.

The AIR Policy Language

AIR (AMORD In RDF) (Kagal, Hanson, and Weitzner 2008) is a policy language which is represented in RDF and provides several classes and properties for defining policies. For a policy in AIR, one can define one or more rules which feature particular patterns, and a certain event is compliant with the policy if it satisfy all the defined patterns. Currently, the AIR Policy Language has been used in the TAMI project (Weitzner et al. 2006) which aims at supporting accountability in large-scale aggregation and inferencing across heterogeneous information systems. An advantage of using AIR is that the reasoner will return a detailed explanation of why access to certain photos is compliant with the policies. The

Figure 1: The Justification UI in Tabulator.

Justification UI extension (see Fig. 1) of Tabulator can be used to provide a clear presentation of the explanation.[1] This allows the user to easily review his policies and see if the rules are properly defined.

Tabulator

The Tabulator (Berners-Lee et al. 2006) is both a browser and an editor of RDF data on the Web. Currently, it can be used both as an extension to the Firefox browser or as a Web application. It allows a user to explore data related to a particular resource on the Web by automatically recognising and following RDF links. It also allows user to edit RDF data in the same interface. While other RDF data browsers such as the OpenLink RDF Browser[2] and Disco[3] exist, Tabulator is chosen in this project because of several of its features. Firstly, Tabulator can be easily extended to provide a customised view of data of a particular type, while the user can still explore the RDF data using the standard views. In addition, Tabulator provides mechanisms for updating the RDF data if it detects that the data sources is editable.

System Design

Our system combines several Web technologies to provide users with a better access control mechanism over their photos. It also aims at increasing the values of tags by representing tagging activities in RDF, such that the access control mechanism can take advantage of the rich information provided by the tags by combining them with the social network information about the users in the Semantic Web.

The main component of our system is the server side script (see Fig. 2) which mediates interactions between the user, the RDF data storage and external services such as OpenID providers and the AIR Reasoner. The system allows user to login using their FOAF URI. The user can create and modify his photo albums, and import metadata of photos from other Web sites where his photos are hosted. Currently our prototype system provides an interface in Tabulator for importing data from Flickr. Of course, importing metadata of photos from other photo sharing sites is possible and users can even host their photos on their own servers. In addition, the user can create new access control policies for his photo albums.

[1] http://dig.csail.mit.edu/TAMI/2008/JustificationUI/howto.html
[2] http://demo.openlinksw.com/rdfbrowser/
[3] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/

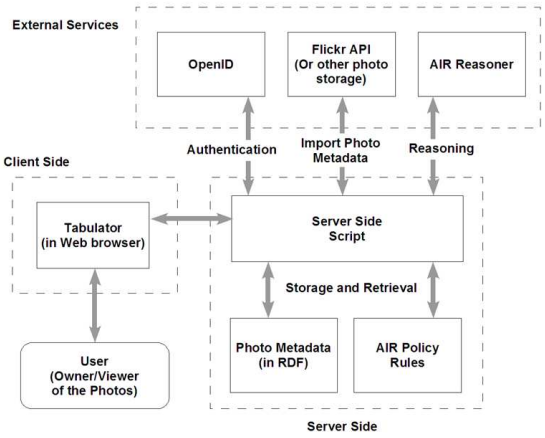

Figure 2: System Architecture.

On the other hand, a user can also browse the photo album of other users. In this case the server side script will gather all relevant data, including the FOAF profile of the owner, the photo album RDF data and the access control policies, and submit them to the AIR Reasoner which determines whether the user is allowed to access the photos. The server side script will then serve RDF data to the user based on the result of the reasoner.

In the following we describe in detail several main components of our proposed system.

OpenID Authentication

Our system relies on the URI of a user to determine whether he should be given access. Hence, the system need to verify whether the user is really in possession of the FOAF file defining his identity. Our system makes use of OpenID to authenticate a user. It assumes that a user has already included his OpenID in his FOAF profile using the `foaf:openid` property. Hence, when a user tries to access photos by providing his URI, the system will go to his FOAF file, obtain his OpenID, and authenticate the user with his OpenID provider. In this way, the system is able to check whether the user is the person identified by the given URI who has control over the corresponding FOAF profile.

Photo Metadata in RDF

Our system relies on an ontology to describe the metadata of the photos and the entities involved in the access control process. Firstly, we create a class called `PhotoAlbum`. An instance of `PhotoAlbum` can be described as containing one or more photos by using the `Contains` property. Using the Tag Ontology, we can describe photos as having one or more taggings which are created by a certain user and are associated with a set of tags. We also create a property called `Owner` which describes the relation between a photo album and a person.

On the other hand, each photo album can be associated with one or more access control policies which are written in the AIR Policy Language. The property `ACPolicy` is created to describe this relation. In addition, a class called `AccessEvent` is created to describe the situation in which a certain user attempts to access the photos in the photo album. An access event involves a person and a photo. Hence, the task of the AIR Reasoner is to determine whether a particular access event is compliant with the access control policies. Fig. 3 gives a summary of the major classes and properties in our ontology.

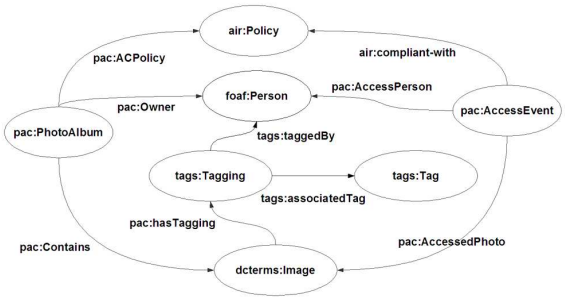

Figure 3: The Photo Access Control Ontology.

Figure 4: The photo displaying pane in Tabulator.

User Interface in Tabulator

We extend Tabulator to provide the user interface to the system. The advantage of using Tabulator as the user interface over developing a new one is that users can explore the RDF data while having a traditional album view of their photos at the same time. In addition, using Tabulator means that the server side script needs only to serve RDF data and does not need to worry about the presentation. Moreover, a customised view of a photo album can be easily done by developing an extension to Tabulator.

Currently, the prototype of our system features a new pane in Tabulator, which will be launched automatically when Tabulator detects that the object being viewed is of the type `PhotoAlbum`. The pane displays photos in the photo album along with the tags assigned to them. Users can filter the photos by checking one or more tags (see Fig. 4). In addition to this, there is also a pane in which users can import photo metadata from Flickr and create access control policies for their photo albums.

Prototype

In our prototype system Tabulator is extended to provide a customised view of a photo album and an interface for importing photo metadata from Flickr into RDF.[4] In the case of using Flickr as a photo storage and our system for access control, a user needs to set his photos private on Flickr, and authenticate himself against Flickr when he creates the RDF metadata. Our system then obtains the absolute URIs of the photos which are made up of several secret IDs.[5] The system can be implemented in a way that the server obtains temporary copies of the photos such that their real URIs will not be revealed to the viewers.

The system now allows a user to provide his FOAF URI and authenticate himself using OpenID. The server side script will then combine his identity with the RDF metadata of the photos and the FOAF profile of their owner, and submit them to the AIR reasoner to determine which photos can be accessed. However, the access control policies have to be written manually for the time being. Providing a graphical user interface for creating access control policies remains a challenging task. At this moment we aim at extending Tabulator to provide an interface for creating simple types of policies, such as those based on the FOAF ontology or involve restricting access to users who are members of a particular organisation.

Use Case

We now present a use case of our system. We use the Notation3 (N3) language (Berners-Lee 2006) when presenting RDF data. A user called Alice has some photos of her birthday party. The photos are stored on Flickr and are all private photos. They are also assigned some tags such as *alice*, *birthday*, *party*. Bob, a friend of Alice, was in Alice's party and he wants to access the photos owned by Alice. We assume that both users have their own OpenID and have properly included this information in their FOAF profile. We also assume that the URIs of Alice and Bob are as follows.

```
http://alice.example.com/foaf.rdf#me
http://bob.example.com/foaf.rdf#me
```

In addition, we assume that Alice has mentioned in her own FOAF profile that Bob is her friend by using the `foaf:knows` property. We also assume both users have installed the Tabulator extension in their browsers.

Importing Metadata of Photos

In order to make use of the access control function of our system, Alice needs to import the metadata of her photos from Flickr. Firstly, she authenticates herself using her OpenID by providing her FOAF profile URI. After receiving her OpenID, the system redirects Alice to the corresponding Web site for authentication. After the authentication process is finished, Alice can create a new album in the system, and choose from her collection in Flickr photos which are to be included in the album. All the tags assigned to the photos will also be identified. The system creates an RDF file storing all the metadata of the photos Alice has imported. Part of the data file in N3 is shown as follows.

```
:pa01 a pac:PhotoAlbum;
  pac:Owner <http://alice.example.com/foaf.rdf#me>;
  pac:Contains <http://flickr.com/secret/1.jpg> .
<http://flickr.com/secret/1.jpg>
  pac:hasTagging :tagging01.
:tagging01
  tags:associatedTag :t_alice, :t_birthday ;
  tags:associatedTag :t_party, :t_bob ;
  tags:taggedBy
    <http://alice.example.com/foaf.rdf#me> .
```

Alice will be given a URI of this photo album, so that she can tell her friends how they can access her photos. Note that the true URL of the RDF file will not be disclosed and that only the server side script should have access right to the file. This is to avoid other users from bypassing the access control mechanism and browse the photos directly. The URI of the above photo album may look something like this:

```
http://www.example.com/photoalbums/alice/pa01
```

Creating Access Control Rules

After importing the metadata, Alice can then specify the access control rules associated with the photo album. In this case, Alice creates a rule as follows: photos which are assigned the tags *alice*, *birthday* and *party* can only be accessed by users who are mentioned in her FOAF profile using the `foaf:knows` property. The system should generate an RDF file which contains the following policy.

```
:R a air:Policy;
  air:rule [
    air:pattern {
      :E pac:AccessPerson :U; pac:AccessedPhoto :P.
      :P pac:hasTagging :T. :T tags:taggedBy :O;
      :T tags:associatedTag :t_birthday, :t_party.
      :T tags:associatedTag :t_alice.
      :O foaf:knows :U . };
    air:description (:E " is compliant with " :R);
    air:assert {:E air:compliant-with :R.}; ].
```

In other words, for any access event involving a user U trying to access a photo P to be compliant with the access control policy R, U has to be "known" by the owner O of P – who is Alice in this case, and the P needs to be one which is assigned the tags *alice*, *birthday* and *party*. The system will attach this policy to the photo album by inserting the following triple into the photo album RDF file.

```
:pa01 pac:ACRules :R .
```

Accessing the Photos

After Alice has finished all the preparations, she can send the URI of the photo album to her friends. Bob receives the

[4] http://dig.csail.mit.edu/2005/ajar/ajaw/Developer.html

[5] See http://www.flickr.com/services/api/misc.urls.html

URI and enters it into the browser. He first goes through the OpenID authentication process by providing his FOAF URI. When the server receives a request of the URI, it actually redirects the browser to the server side script with the owner of the photo album (Alice) and the name of the photo album (pa01) as input parameters. The server side script retrieves the RDF file containing the photo album and checks if there are any associated access control rules. It then constructs a temporary RDF file by combining the owner's FOAF profile, the metadata of the photos, and some triples created to describe the instances of `AccessEvent`. For example, in this scenario the following triples will be created.

```
:AccessEvent1 a pac:AccessEvent;
  pac:AccessPerson
    <http://bob.example.com/foaf.rdf#me>;
  pac:AccessedPhoto
    <http://farm1.static.flickr.com/secret/01.jpg>.
```

The server side script then forwards the URLs of this file and the file containing the policies to the AIR Reasoner. The reasoner returns a response coded in N3 which indicates which of the access events are compliant with the access control rules. For example, the response from the reasoner in this case looks something like the following, as Alice has mentioned in her FOAF profile that she knows Bob.

```
:AccessEvent1 air:compliant-with :R .
```

The server side script, on receiving this response, creates a new instance of `PhotoAlbum`, attaches to it photos with can be accessed by Bob, and then sends the data in RDF format to the browser. Finally, Tabulator in the browser reads the data and presents Bob with the photos he is allowed to see. The complete set of files and the actual result returned by the reasoner are available by accessing our project page.[6]

Discussions

While the above scenario features a rather simple usage of the proposed system, it does highlight some of its advantages. Firstly, by including the tags assigned to the photos as part of an access control policy, the system allows users to create policies based on what the photos are about. In existing photo sharing sites, users can only specify whether an individual photo or a set of photo is visible to a certain group of users, and this has nothing to do with what the photos are about. On the other hand, in our system users can create meaningful policies such as that photos with the tags *conference* and *presentation* can be accessed by one's colleagues, and that photos with the tags *graduation* and *ceremony* can be accessed by students from the same university.

In addition, by representing tags in RDF, the system is able to connect user-created tags with other linked data on the Web, thus enriching the semantics of the tags. For example, by allowing users to bind tags to classes in some ontologies, policies which involves different properties of the tags can be created. Consider a user tagging photos of his trip in England with tags such as *london* and *manchester*. If he specifies that the two tags are the same as the instances `London` and `Manchester` in the DBpedia ontology (http://dbpedia.org/) using the `owl:sameAs` property, he can then create a single policy which controls access to his photos of cities in England, as the reasoner will be able to infer that London and Manchester are cities in England using the properties available in DBpedia.

The system also allows users to specify the group of people who can access the photos by referring to linked data on the Web. While FOAF only provides a rudimentary relation (`foaf:knows`) for describing friendship, it can easily be extended to describe more fine-grained relationships. In addition to FOAF, a user can also create policies which grant access to, for example, students taking a certain class, members of a research group, participants of an academic conference, or even a combination of the above examples. In this way, users can rely on the corresponding organisations they trust to maintain the list of members, and avoid enumerating the names of all people in an access control list.

Related Work

Access control to online resources forms part of the broader notion of policy (Bonatti et al. 2006) on the Web. Various access control mechanisms to private resources on the Web have been discussed by different authors. For example, PeerTrust (Gavriloaie et al. 2004) is proposed to provide an access control mechanism based on semantic annotation, policies and automated trust negotiation. Users involve in an iterative process of exchanging credentials to establish trust between them. While the system provides a reliable mechanism to establish trust between two parties, the process and the need of providing certain private information involve substantial overhead which may not be necessary in common online activities such as photo sharing.

Lalana et al. (2006) propose a Web-based policy management framework called Rein which makes use of Semantic Web technologies. The framework provides an ontology which can be used to describe what access control policies are attached to a user's resources and in what languages the policies are written. The authors also describe a scenario similar to our use case in which a user restricts access to her photos by creating a policy using FOAF and the Rein Framework. As the system proposed in this paper relies only on policies written in AIR, the ability to support policies written in different languages is an interesting feature which we would like to look into in the future.

Tootoonchian et al. (2008) introduce an access control scheme called Lockr. It allows a user to send to other users something called social attestations which they use to prove their social relationship with the user to any Web sites storing personal contents. Lockr also allows policies based on different kinds of social relationship to be created by allowing a user to construct a social access control list. The authors propose that information about a social network should be separated from content delivery. However, the possibility of using linked data on the Web is not considered.

In addition, Yagüe et al. (2003) describes a layered access control scheme based on Semantic Web technologies

[6] http://people.csail.mit.edu/albert08/aaai/index.html

and gives access to users based on their as well as the resources' semantic properties. Working in the context of Semantic Web services, Agarwal and Sprick (2004) studies how access control policies to a composite Web service can be determined based on those of its components. While these studies focus on the utilising Semantic Web technologies, they do not discuss how the potentiality of linked data can be exploited to help create expressive policies.

Conclusions

We propose a access control system based on a decentralised authentication protocol, descriptive tags and linked data of social networks in the Semantic Web. It allows users to create expressive policies for their photos stored in one or more photo sharing sites, and users can specify access control rules based on open linked data provided by other parties. While we focus on photo sharing in this paper, the system can easily be extended to provide access control to any other resources on the Web such as bookmarks and videos. From this point onwards, we plan to construct a prototype system which incorporates all the functions mentioned in this paper. We will also investigate how to allow users to add semantics to their tags and create more expressive policies by referring to external data sources such as DBpedia.

References

Agarwal, S., and Sprick, B. 2004. Access control for semantic web services. In *ICWS '04: Proceedings of the IEEE International Conference on Web Services*, 770. Washington, DC, USA: IEEE Computer Society.

Berners-Lee, T.; Chen, Y.; Chilton, L.; Connolly, D.; Dhanaraj, R.; Hollenbach, J.; Lerer, A.; and Sheets, D. 2006. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop*.

Berners-Lee, T. 2006. Notation3 (N3) a readable rdf syntax. *http://www.w3.org/DesignIssues/Notation3.html*.

Bizer, C.; Heath, T.; Idehen, K.; and Berners-Lee, T. 2008. Linked data on the web. In Bizer, C.; Heath, T.; Idehen, K.; and Berners-Lee, T., eds., *Proceedings of the Linked Data on the Web Workshop, Beijing, China, April 22, 2008*, CEUR Workshop Proceedings. CEUR-WS.org.

Bonatti, P. A.; Duma, C.; Fuchs, N. E.; Nejdl, W.; Olmedilla, D.; Peer, J.; and Shahmehri, N. 2006. Semantic web policies - a discussion of requirements and research issues. In *Proceedings of 3rd European Semantic Web Conference, Budva, Montenegro, 11th–14th June 2006*, volume 4011 of *LNCS*, 712–724.

Breslin, J. G.; Harth, A.; Bojars, U.; and Decker, S. 2005. Towards semantically-interlinked online communities. In *Proceedings of the Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 - June 1, 2005*, volume 3532 of *LNCS*, 500–514. Springer.

Brickley, D., and Miller, L. 2007. FOAF vocabulary specification, *http://xmlns.com/foaf/spec/*.

Gavriloaie, R.; Nejdl, W.; Olmedilla, D.; Seamons, K. E.; and Winslett, M. 2004. No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In *Proceedings of the First European Semantic Web Symposium, Heraklion, Crete, Greece, May 10-12, 2004*, volume 3053 of *LNCS*, 342–356. Springer.

Gruber, T. 2007. Folksonomy of ontology: A mash-up of apples and oranges. *International Journal on Semantic Web and Information Systems* 3(2).

Kagal, L.; Berners-Lee, T.; Connolly, D.; and Weitzner, D. J. 2006. Using semantic web technologies for policy management on the web. In *Proceedings of The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*. AAAI Press.

Kagal, L.; Hanson, C.; and Weitzner, D. J. 2008. Using dependency tracking to provide explanations for policy management. In *Proceedings of the 9th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2008), 2-4 June 2008, Palisades, New York, USA*, 54–61. IEEE Computer Society.

Kim, H. L.; Yang, S.-K.; Song, S.-J.; Breslin, J. G.; and Kim, H.-G. 2007. Tag mediated society with scot ontology. In *Proceedings of the Semantic Web Challenge 2007 co-located with ISWC 2007 + ASWC 2007, Busan, Korea, November 13, 2007*. CEUR-WS.org.

Kim, H. L.; Scerri, S.; Breslin, J.; Decker, S.; and Kim, H. G. 2008. The state of the art in tag ontologies: A semantic model for tagging and folksonomies. In *Proceedings of the International Conference on Dublin Core and Metadata Applications, Berlin, Germany*.

Miller, A. D., and Edwards, W. K. 2007. Give and take: a study of consumer photo-sharing culture and practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 28 April - 3 May 2007, San Jose, California, USA*, 347–356. New York, NY, USA: ACM.

Newman, R. 2005. Tag ontology design. *http://www.holygoat.co.uk/projects/tags/*.

Recordon, D., and Fitzpatrick, B. 2006. OpenID authentication 1.1, *http://openid.net/specs/openid-authentication-1_1.html*.

Tootoonchian, A.; Gollu, K. K.; Saroiu, S.; Ganjali, Y.; and Wolman, A. 2008. Lockr: social access control for web 2.0. In *WOSP '08: Proceedings of the first workshop on Online social networks, Seattle, WA, USA, August 18, 2008*, 43–48. New York, NY, USA: ACM.

Weitzner, D.; Abelson, H.; Berners-Lee, T.; Hanson, C.; Hendler, J.; Kagal, L.; McGuinness, D.; Sussman, G.; and Waterman, K. K. 2006. Transparent accountable data mining: New strategies for privacy protection. Technical report, CSAIL, Massachusetts Institute of Technology.

Yagüe, M. I.; Antonio Ma n.; López, J.; and Troya, J. M. 2003. Applying the semantic web layers to access control. In *DEXA '03: Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, 622. Washington, DC, USA: IEEE Computer Society.