A Privacy Preference Ontology (PPO) for Linked Data*

Owen Sacco and Alexandre Passant
Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
firstname.lastname@deri.org

ABSTRACT

Linked Data enables people to access other users' data stored in several places, distributed across the Web. Current Linked Data mechanisms mostly provide an open environment where all data is freely accessible, which could discourage some people to provide sensitive data in the Linking Open Data (LOD) cloud. Although the existing Web Access Control (WAC) vocabulary restricts RDF documents to specified users, it does not provide fine-grained privacy measures which specify complex restrictions to access the data. In this paper, we propose a lightweight vocabulary — on top of WAC — called the Privacy Preference Ontology (PPO) that enables users to create fine-grained privacy preferences for their data. The vocabulary is designed to restrict any resource to certain attributes which a requester must satisfy.

Categories and Subject Descriptors

1.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Method; K.4.1 [**Public Policy Issues**]: Privacy; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Design, Security

Keywords

Privacy, Linked Data, WebID, Web Access Control, FOAF, RDF, Named Graphs

1. INTRODUCTION

The Linked Data community encourages Web users to format their data and publish it on the Web in machine processable formats so that other datasets can be linked to this published data. However, as pointed out in [2], one of the challenges of Linked Data is privacy. Datasets are being published in the Linking Open Data (LOD) cloud without any metadata that describes privacy restrictions, and therefore the data is publicly accessible. A vocabulary that describes access control privileges is the Web Access Control (WAC) vocabulary[1]. This vocabulary enables owners to create access control lists (ACL) that specify access privileges to the users that can access the data. The WAC vocabulary defines the `Read` and `Write` access control privileges (for reading or updating data) as well as the `Control` privilege to grant access to modify the ACL. This vocabulary is designed to specify access control to the full RDF document rather than specifying access control properties to specific data contained within the RDF document.

In [13], the authors discuss the importance that protecting data does not only mean granting full access or not, but in certain instances fine-grained access control mechanisms are required to restrict pieces of information. For instance users could define which specific microblog posts in SMOB[2] are shared to certain users only based on `#tags`. Therefore, the Linked Data infrastructure currently lacks mechanisms for creating fine-grained privacy preferences that define which data can be accessed by whom. This might discourage Web users to publish sensitive data such as user's personal information contained in FOAF profiles.

In this paper, we propose the Privacy Preference Ontology (OPO), a lightweight vocabulary on top of the Web Access Control ontology aiming at providing users with means to define fine-grained privacy preferences for restricting (or granting) access specific RDF data. As we rely on Semantic Web technologies to enable these privacy preferences, our proposed vocabulary is platform independent and can thus be used by any system relying on these technologies.

The remainder of the paper is as follows: Section 2 presents some use cases where privacy is a concern. In Section 3, we present our Privacy Preference Ontology (PPO), online at – `http://vocab.deri.ie/ppo#`, and discuss how to apply it to protect sensible data. Section 4 provides a brief description of current privacy research related to protecting RDF data and social networks. Finally, Section 5 concludes this paper and gives an overview of future work.

2. MOTIVATIONS

Open social networks can contain user's information described in RDF, using common vocabularies such as FOAF to describe this data. Applications are being developed to export user information stored within closed social networks into RDF, while various projects now directly support these models to represent user data, such as Drupal 7[3]. Current

*This work is funded by the Science Foundation Ireland under grant number SFI/08/CE/I1380 (Líon 2), by an IRCSET scholarship and by a Google Research Award.

LDOW2011 March 29, 2011, Hyderabad, India.

[1]WAC — `http://www.w3.org/ns/auth/acl`
[2]SMOB — `http://smob.me`
[3]Drupal — `http://drupal.org/`

social networks provide minimum privacy settings such as granting privileges to all people belonging to one's social graph to access her/his information. Imagine a social network where users would be able to specify which information can be shared only to some contacts or friends, e.g. the ones having similar interests. This would make users feel more confident when publishing such information without being concerned that it could be reused. Moreover, such a system will let users fully-control who can access their personal information and who can access their published RDF data. Ideally, data owners can specify a set of attributes which requesters must satisfy in order to be granted access to the requested information. For example a user can set a privacy preference to share an e-mail address only to those who are belonging to his company. This could be achieved by executing a SPARQL query combining a privacy preference pattern and the FOAF description of the requester as suggested in [14]. In this social network scenario, the WebID protocol [15] can be used to authenticate a user and also it provides a secure connection to a user's personal information stored in a FOAF profile [7]. Therefore, once a user authenticates using WebID when visiting another user's profile, the privacy preferences could be checked to determine which information can be accessed.

Another scenario relates to online publications, and in particular microblogging. Currently, most microblogging systems allow any user to access posts created by others. As pointed out in [13], sensitive posts such as the ones shared within an organisation, require more complex access restriction. In SMOB [12], microblog posts are described in RDF using ontologies such as SIOC (for describing posts) and FOAF (for describing user profiles). Additionally, SMOB provides the ability to tag microblog posts with concepts taken from databases such as DBpedia[4] and GeoNames[5] unlike microblogging systems such as Twitter that only allow text-based tags. While it relied on the Online Presence Ontology (OPO) [14] so that messages can be directed to particular users, further privacy preferences are required such as restricting access to posts only to some people, for example the ones having interests related to the post's topic (based on its tags). Since SMOB relies on Semantic Web technologies and Linked Data, advanced privacy preferences can be easily applied. For example, if a user wants to restrict a microblog post tagged with a particular topic to a group of friends, this privacy preference can be applied by restricting the post to users being interested in one of the "semantic tag" used in the post, this tag being defined with its own URI, e.g. from DBpedia.

3. PRIVACY PREFERENCE ONTOLOGY

The previous use cases illustrate situations where fine-grained privacy preferences are required. We therefore created a dedicated vocabulary called the Privacy Preference Ontology (PPO) to describe privacy preferences that can restrict access to information represented as Linked Data. Since Linked Data uses RDF as a representation format, this requires the privacy preferences to restrict access to particular RDF data. In particular, the vocabulary should provide the ability to restrict access to: (1) a particular statement; or (2) to a group of statements (i.e. an RDF graph); or (3) to a resource, either as a subject or an object of a particular statement.

[4]DBpedia — `http://dbpedia.org/`
[5]GeoNames — `http://www.geonames.org/`

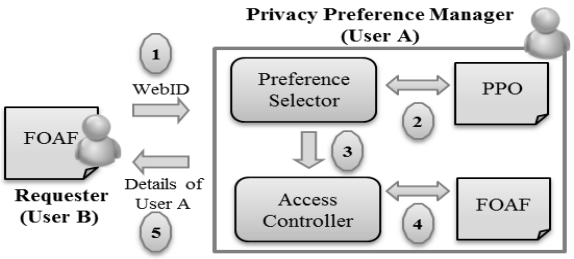

Figure 2: The Privacy Preference Manager

Access are restricted according to patterns which users (that want to access data) must satisfy, for instance having a particular interest or being a member of a group. We rely on the Web Access Control vocabulary to describe the access privilege to the data: either `Read`, `Write` or both. Therefore, a privacy preference contains properties defining: (1) which resource, statement or graph to restrict access; (2) the type of restriction; (3) the access control type; and (4) a SPARQL query containing a graph pattern representing what must be satisfied by the user requesting information.

Currently we assume that the user's information is trustworthy. Eventually, we plan to extend the vocabulary to cater for situations where the user's information is not a reliable source, by relying on trust measures to do so [6].

One way to use this ontology is to define a personal Privacy Preference Manager (PPM), providing users with means to specify preferences based on their FOAF profile. The PPM can then be used to grant privileges to requesters that want to access the user's information. Figure 2 illustrates the related concept: (1) a requester authenticates to the other user's PPM using the WebID protocol; (2) the privacy preferences are queried to identify which preference applies; (3) the preferences are matched according to the requester's profile to test what the requester can access; (4) the requested information (in this case, FOAF data) is retrieved based on what can be accessed; and (5) the requester is provided with the data she/he can access. This privacy manager will not be limited to only data described in FOAF, but to any RDF data since PPO is ontology-agnostic. For instance, it can be used to restrict microblog posts described using SIOC and other ontologies used in SMOB.

3.1 Ontology

The Privacy Preference Ontology (PPO) illustrated in figure 1 provides: (1) a main class called `PrivacyPreference` that defines a privacy preference; (2) some properties to define which statement, resource and/or graph is to be restricted; (3) some properties that define conditions in order to create specific privacy preferences; (4) some properties to define which access privilege should be granted; (5) and some properties that define which attribute patterns a requester must satisfy. Moreover, a user may want to define global preferences such as restricting access to values that have a specific property. For instance, if one wants to restrict access to all statements containing `foaf:homepage`, rather than only the ones linking to a specific homepage, s/he can create a condition that restricts every statement containing the `foaf:homepage` property. Hence, the restriction levels

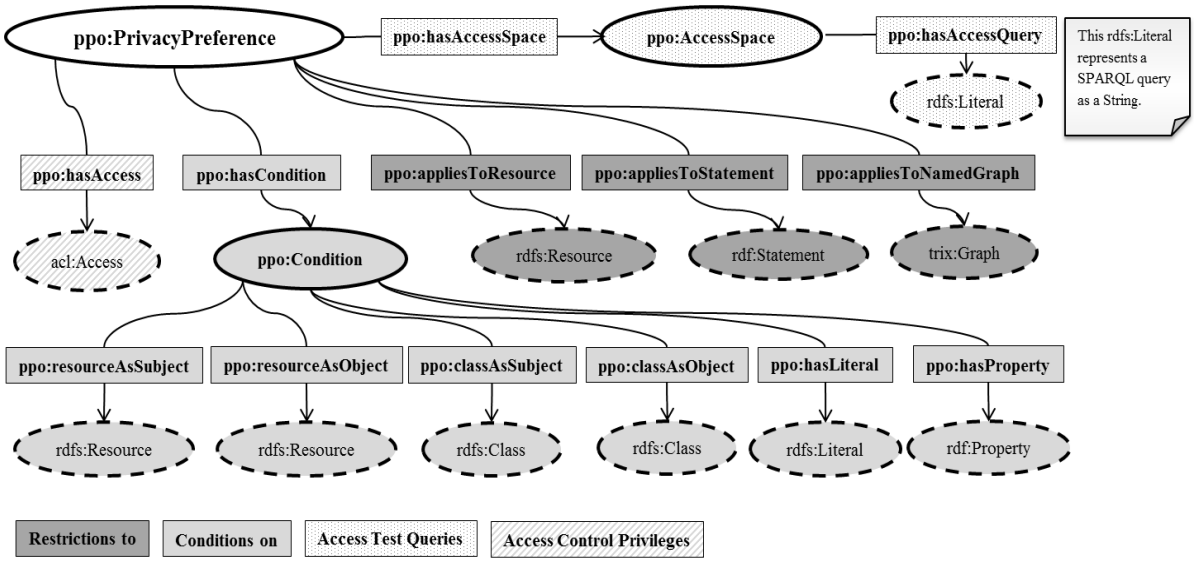

Figure 1: The Privacy Preference Ontology

provided by PPO can be seen as a tree graph that contains at the top node an instance of a class or property from any ontology down to specific data value nodes found in RDF statements. The privacy preference restrictions can be applied to any node within this tree graph. The other classes and properties provided by PPO are explained below.

Condition. The `Condition` class is used to define restrictions within a privacy preference. These restrictions can be applied using the properties provided by this class. The `resourceAsSubject` property provides a condition whereby a resource is used as a subject in a statement. Similarly, the `resourceAsObject` property is used to apply a condition whenever a resource is defined as an object. In certain cases, users would want to specify instances of a particular class. This is achieved by using the `classAsSubject` or the `classAsObject` properties. When the `classAsSubject` property is used, the privacy preference applies to those statements that contain the instance of the class specified as the subject of the statement. Additionally, if the `classAsObject` property is used, then the privacy preference applies to those statements that the object defines the instance of the class. The property `hasProperty` restricts all instances of a particular property used in RDF statements. This means that if one is using `hasProperty` together with `foaf:phone`, all statements containing this property will be restricted. In certain scenarios, users would require to restrict access to statements based on a particular literal value contained within statements. This can be achieved by using the `hasLiteral` property. This property is useful when the user is not aware of which property describes the literal. In this scenario, the `hasLiteral` property must be used with care since if there is another statement with the same value but has a different property, then this statement with a different property is also restricted. This property can also be used together with the `hasProperty`. For instance if a user wants to restrict a particular value of a specific property, then both the `hasProperty` and the `hasLiteral` must be used, such as restricting access to a mobile phone number (`foaf:phone`) but allowing access to the land-line phone number, based on the number prefix (`hasLiteral`).

appliesToResource. The `appliesToResource` property is used to specify which resource must be restricted. This property restricts statements that contain the resource's URI both when it is a subject or an object. The user may create a condition to distinguish when the resource is either a subject or object by using the `resourceAsSubject` or `resourceAsObject` properties respectively.

appliesToStatement. The `appliesToStatement` property is used to specify which statement must be restricted. When the user uses this property, the user must specify the *subject*, *predicate* and *object* of the statement which needs to be restricted.

appliesToNamedGraph. In certain cases, users require a group of statements to be restricted using similar conditions. Yet, it would be cumbersome to create a preference for each statement using the `appliesToStatement` property. Hence, users can use named graphs [1] to combine statements and apply a privacy preference to the graph, using the `appliesToNamedGraph` property. Named graphs are identified with URIs which can be used to refer to a particular named graph that needs to be restricted. Although named graphs are not yet standardised within the RDF specification these are accepted by the SPARQL specification and are in the scope of the new W3C RDF Working Group.

hasAccessSpace. In the previous scenarios, we mentioned that it may be cumbersome for users to update their preferences by adding or removing users manually, since user's interests or relationships change over time. Rather than specifying who can access the resources, we suggest to use a set of attributes specifying which ones are required to access some data. This can be done by using a SPARQL `ASK` query that contains a graph pattern specifying which attributes and properties must be satisfied. By executing the query on the requester's FOAF profile, we know whether the requester satisfies these attributes or not. The SPARQL query is described as a `Literal` in the privacy preferences using the `hasAccessQuery` property. The `hasAccessQuery` property is defined within a class called `AccessSpace` which

denotes a space of access test queries. Finally, the property `hasAccessSpace` represents the relationship between the privacy preference and the access space. Unfortunately, the current SPARQL specification does not cater for `triggers` similar to the DBMS `trigger` concept[6]. Therefore, the query defined in `hasAccessQuery` has to be executed by a manual system call rather than called automatic if a `hasAccessQuery` property appears within the privacy preference.

hasAccess. The Privacy Preference Ontology provides a property that describes the type of access control to be granted when a privacy preference applies. The `hasAccess` property defines the access control described using the Web Access Control vocabulary described in section 1.

3.2 Creating Privacy Preferences

Privacy preferences can easily be created using the PPO and the Web Access Control vocabulary. For example if a user wants to create a privacy preference that restricts the phone number to whoever works at DERI, the following has to be defined[7].

```
<http://www.example.org/pp1>
  a ppo:PrivacyPreference;

  ppo:hasCondition
    [
      ppo:hasProperty foaf:phone
    ];

  ppo:hasAccess acl:Read;

  ppo:hasAccessSpace
    [
      ppo:hasAccessQuery
      "ASK {
        ?x foaf:workplaceHomepage
          <http://www.deri.ie> }"
    ].
```

This example illustrates that wherever in the user's profile there is a statement that contains a property `foaf:phone` then all statements containing this property are restricted. If the user requires a particular `foaf:phone` to be restricted, then the user must also define the phone number in the condition by using the `hasLiteral` property. As mentioned in the previous section, the SPARQL query is executed on the requester's FOAF profile by the system once it parses that there is a query. The query returns either True or False whether the requester's information satisfies the graph pattern or not since the query is a SPARQL `ASK` query. If the query returns a `Yes` then the requester is granted access to the statement, otherwise the requester is not allowed access[8].

The following example shows how to restrict a microblog post to users that share an interest similar to the concept used to tag the post. Restricting posts tagged with the concept of Linked Data to all users interested in Linked Data is done as follows:

```
<http://www.example.org/pp2>
  a ppo:PrivacyPreference;

  ppo:appliesToResource <http://smob.me/
      user/xyz/post1>;
  ppo:assignAccess acl:Read;

  ppo:hasCondition [
        ppo:hasProperty tag:Tag;
        ppo:resourceAsObject
          <http://dbpedia.org/resource/
              Linked_Data> ];

  ppo:hasAccessSpace [
      ppo:hasAccessQuery
      "ASK {
        ?x foaf:topic_interest
          <http://dbpedia.org/resource/
              Linked_Data> }" ].
```

4. RELATED WORK

The Platform for Privacy Preferences (P3P)[9] specifies a protocol that enables Web sites to share their privacy policies with Web users. The privacy policies are expressed in XML which can be easily parsed by user agents. This platform does not ensure that Web sites act according to their publicised policies. Moreover, since this platform aims to enable Web sites to define their privacy policies, it does not solve our aim of enabling users to define their own privacy preferences. The Protocol for Web Description Resources (POWDER)[10] is designed to express statements that describe what a collection of RDF statements are about. The descriptions expressed using this protocol are text based and therefore do not contain any semantics that can define what the description states. Therefore, our approach enables users to define what the privacy preferences are about and hence facilitate other systems to use such preferences.

The authors in [9] propose a privacy preference formal model consisting of relationships between objects and subjects. Objects consist of resources and actions, whereas subjects are those roles that are allowed to perform the action on the resource. The privacy settings based on this formal model are implemented using Protune [3], a policy framework that consists of a policy language and a policy reasoner. This implies that any system using this method must have the Protune framework. Since our aim is to propose a light weight vocabulary that can be platform independent, therefore this approach of using the Protune policy engine does not solve our goal. Moreover, the proposed formal model relies on specifying precisely who can access the resource. Our approach provides a more flexible solution which requires the user to specify attributes which the requester must satisfy. The authors in [4] propose an access control framework for Social Networks by specifying privacy rules using the Semantic Web Rule Language (SWRL) [11]. This approach is also based on specifying who can access which resource. Moreover, this approach relies that the system contains a SWRL reasoner. In [5] the authors propose a relational based access control model called `RelBac` which provides a formal model based on relationships amongst communities and resources.

[6]However, some SPARQL engines provide triggers, such as ARC2 — `http://arc.semsol.org`

[7]We assume that a PPO interpreter would know the common prefixes for SPARQL queries, while they could also be defined in the ASK pattern.

[8]As previously mentioned, so far, we assume that we can trust the statements defined in the requester FOAF file, and we tackle this issue separately.

[9]P3P — `http://www.w3.org/TR/P3P/`

[10]POWDER — `http://www.w3.org/TR/powder-dr/`

[11]SWRL — `http://www.w3.org/Submission/SWRL/`

This approach also requires to specifically define who can access the resource(s).

The authors in [11] propose a tag-based model to create privacy settings for medical applications that consist of annotating resources with different access policy rules. The privacy rules are denoted in a system specific language which only the system can interpret the access control. The authors in [10] also propose an annotation based access control model. This approach enables users to annotate the resource and also to annotate users. The access control rules therefore specify which resource annotations can be accessed by which user annotations. Although this approach might be more flexible than other systems, it still relies on specifying who can access the resource.

In [14] the authors propose a method to direct messages, such as microblog posts in SMOB, to specific users according to their online status. The authors also propose the idea of a `SharingSpace` which represents the persons or group of persons who can access the messages. The authors also describe that a `SharingSpace` can be a dynamic group constructed using a SPARQL `CONSTRUCT` query. However, the proposed ontology only allows relating the messages to a pre-constructed group.

In [8] the authors propose a system whereby users can set access control to RDF documents. The access controls are described using the Web Access Control vocabulary by specifying who can access which RDF document. Authentication to this system is achieved using the WebID protocol [15]. This protocol uses FOAF+SSL techniques whereby a user provides a certificate which contains a URL that denotes the user's FOAF profile. The public key from the FOAF profile and the public key contained in the certificate which the user provides are matched to allow or disallow access. Our approach extends the Web Access Control vocabulary to provide more fine-grained access control to the data rather than to the whole RDF document.

5. CONCLUSION AND FUTURE WORK

In this paper we argue that there are not sufficient fine-grained privacy preferences for Linked Data. We therefore proposed a light weight vocabulary which provides classes and properties to define fine-grained privacy preferences for RDF data. The privacy preferences define what needs to be protected, the conditions to create fine-grained restrictions; which access control privilege will be granted and a space to define which attributes a requester must satisfy in order to access the resource. The access control privileges are described using the Web Access Control vocabulary. We plan to extend the PPO to also restrict actions which are commonly found in Social Web applications and we also plan to extend our work to cater for conflicting privacy preferences.

Additionally, we will investigate a formal model for PPO and its relationships with RDFS and OWL entailments, to ensure that preferences can also apply to inferred data (for example to restrict the sub-properties or subclasses of the property or class being restricted). This step will also be required to be sure that will not be any vulnerability attacks caused by inferred statements. Moreover, we are currently developing the Privacy Preference Manager mentioned in section 3 which provides a user-friendly interface where users can specify privacy preferences described using the Privacy Preference Ontology, as well as applying the privacy preferences when accessing the RDF data.

6. REFERENCES

[1] C. Bizer and J. Carroll. Modelling Context Using Named Graphs. In *W3C Semantic Web Interest Group Meeting*, 2004.

[2] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 2009.

[3] P. Bonatti and D. Olmedilla. Driving and Monitoring Provisional Trust Negotiation with Metapolicies. In *Sixth IEEE International Workshop on Policies for Distributed Systems and Networks*, 2005.

[4] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. A Semantic Web Based Framework for Social Network Access Control. In *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, SACMAT '09*, 2009.

[5] F. Giunchiglia, R. Zhang, and B. Crispo. Ontology Driven Community Access Control. *Trust and Privacy on the Social and Semantic Web, SPOT'09*, 2009.

[6] O. Hartig. Querying trust in rdf data with tsparql. In *6th Annual European Semantic Web Conference, ESWC'09*, 2009.

[7] B. Heitmann, J. Kim, A. Passant, C. Hayes, and H. Kim. An Architecture for Privacy-Enabled User Profile Portability on the Web of Data. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec '10*, 2010.

[8] J. Hollenbach and J. Presbrey. Using RDF Metadata to Enable Access Control on the Social Semantic Web. In *Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge, CK'09*, 2009.

[9] P. Kärger and W. Siberski. Guarding a Walled Garden Semantic Privacy Preferences for the Social Web. *The Semantic Web: Research and Applications*, 2010.

[10] P. Nasirifard, V. Peristeras, and S. Decker. Annotation-Based Access Control for Collaborative Information Spaces. *Computers in Human Behavior*, 2010.

[11] S. Nepal, J. Zic, F. Jaccard, and G. Kraehenbuehl. A Tag-Based Data Model for Privacy-Preserving Medical Applications. *Current Trends in Database Technology*, 2006.

[12] A. Passant, J. Breslin, and S. Decker. Rethinking Microblogging: Open, Distributed, Semantic. In *Proceedings of the 10th International Conference on Web Engineering, ICWE'10*, 2010.

[13] A. Passant, P. Kärger, M. Hausenblas, D. Olmedilla, A. Polleres, and S. Decker. Enabling Trust and Privacy on the Social Web. In *W3C Workshop on the Future of Social Networking)*, 2009.

[14] M. Stankovic, A. Passant, and P. Laublet. Directing status messages to their audience in online communities. In *Proceedings of the 5th International Conference on Coordination, Organizations, Institutions, and Norms in Agent Systems*, 2010.

[15] H. Story, B. Harbulot, I. Jacobi, and M. Jones. FOAF + SSL : RESTful Authentication for the Social Web. *Semantic Web Conference*, 2009.