Semantics-aware Privacy and Access Control: Motivation and Preliminary Results

E. Damiani, S. De Capitani di Vimercati, C. Fugazza, and P. Samarati
{damiani,decapita,fugazza,samarati}@dti.unimi.it

Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
via Bramante 65, Crema

Abstract

Semantic Web languages like OWL and RDFS promise to be viable means for representing metadata describing users and resources available over the Internet. Recently, interest has been raised on the use of such languages to represent individual data items contained in Personally Identifiable Information *(PII), supporting fine-grained release. To achieve this goal, the informative content of a credential must be dissected into atomic components so that users can selectively single out those to be released. In this paper, we outline three different methodologies for taking advantage of fine-grained personal information for controlled release at both policy writing and evaluation time, and pinpoint aspects that should be considered for an effective exchange and evaluation of policies.*

1. Introduction

Nowadays the World Wide Web reaches the widest audience ever conceived through a broad range of devices such as computers, phones, and PDAs. Security and privacy concerns are increasingly important in this environment, where controlling the release, retention, and secondary use of personal data have become key issues. While encryption-based technologies such as the Public Key Infrastructure [11] guarantee credentials' unforgeability, a framework for empowering the user with full control over information release during the exchange of certificates on the Web is still missing [5, 10]. Key requirements for this framework include:

1. A data model for representing credential information and a language enabling:
   - end users to state and apply policies expressing their preferences on the disclosure and acceptable secondary use of personal data;
   - service providers to dynamically define the requirements to be met by clients.
2. A decision mechanism enabling uniform evaluation and enforcement of policies.

Advanced modeling of *Personally Identifiable Information* (PII) allows controlling its release according to users' privacy requirements. The *Platform for Privacy Preferences* (P3P) [9] is an XML-based standard language for expressing data-collection and data-use practices in a standard format. The W3C has also proposed APPEL (*A P3P Preference Exchange Language*) [1] to allow users to specify their privacy preferences. Joint use of P3P and APPEL should enable comparing client's privacy preferences with the data collection practice of a server, deciding whether a transaction can be carried out or should be aborted.

Here, we shall focus on P3P *data schema*, that provides us with a well understood type-space for the definition of the data items that can be exchanged in a client-server interaction. Unfortunately, P3P data schemata still lack the expressive power and the clearly defined semantics required for the definition of complex user credentials (a preliminary assessment of recent work on this issue is presented in [16]). Semantic Web languages like OWL [14] and RDFS [12] lend themselves very well to advanced representation of personal information inasmuch they allow for defining cross-cutting relationships linking semantically equivalent data items (e.g., birth dates) appearing in multiple credentials (e.g., a passport and a driver license). In our previous work [4] we showed how the expressive power of standard XML-based access control languages can be increased to take advantage of ontology-based descriptions of the resources to be protected. Here, we address the problem of using ontology to increase the expressive power of P3P data schema. Specifically, we present some techniques allowing for the informative content of a user credential to be decomposed into atomic components, so that users can non-ambiguously single out items to be released. The remainder of this paper is structured as follows. Section 2 defines an ontology-based abstract model underlying P3P data schema. Section 3 outlines how this model can be used to increase the expressive power of the language, achieving full control over the disclosure of PII during client-server interactions. Section 4 discusses how our model can be translated into Semantic Web-style metadata. Finally, Section 5 draws the conclusions.

2. The role of P3P

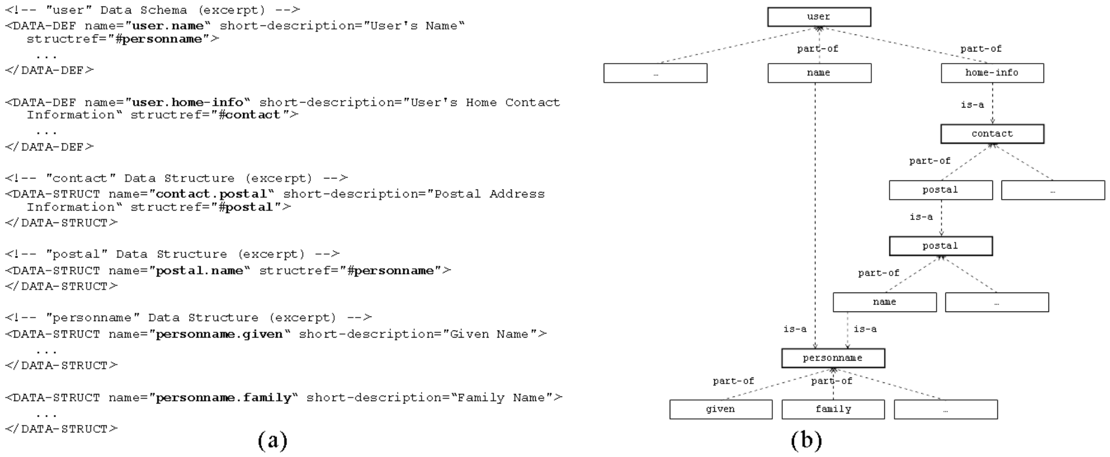
The core concept of P3P data schema is *data element* representing a single data item that can be either a root (unstructured) value or a more complex *data structure* composed of a set of data elements. As an example, consider the definition of the `personname` data structure in the P3P syntax illustrated in Figure 1(a). Individual elements of a structure are linked to one or more *categories* selected from a flat list of mutually exclusive identifiers. Even if the semantics of categories could be extended, for instance allowing generalization and specialization of concepts, by the state of the art they represent just an alternative, unstructured categorization of data and are not considered in our model for the sake of conciseness.

P3P data elements (and structures) are grouped into four *data element sets* (`user`, `thirdparty`, `business`, and `dynamic`). Data element sets have been introduced for rooting the overall containment structure of P3P credentials. In other words, data element sets are data structures (i.e., they point to data elements) that cannot be included within other structures. Since each data element can appear in more than one structure, the overall containment relation is easily seen to be a semi-lattice rather than a forest. Figure 1 shows a portion of P3P base data schema definition clearly showing the semi-lattice structure with the multiple references to the `personname` data structure. This can also be seen in the definitions of the `user`, `contact`, and `postal` data structures.

The class diagram in Figure 1(b) shows P3P data elements referencing the structure enclosing them via a *part-of* relation and the structure defining them via a *is-a* relation. By collapsing the *is-a* relation, the containment semi-lattice structure becomes evident. In the figure, we use different kinds of boxes to discriminate among the different entities. In particular, we use thick boxes for data structures (e.g., `contact`) and thin boxes for data elements (e.g., `name`). Figure 2 illustrates the same structure in the formalism adopted in our work, where data structures are presented as classes and the data elements composing them as attributes. To help disentangling the different nature of entities, here we capitalize class names (using CamelCase for composite names) and stick to the original hyphenated syntax of P3P for attributes. This way the `Postal` data structure can be distinguished from the `postal` data element of the `Contact` data structure.

2.1. An abstract model for P3P

We are now ready to provide a structural definition of P3P semantics by arranging in a single structure the data element sets and the data structures composing them (such as `date`, `login`, and `http-info`), down to the single data elements and the categories grouping them (e.g., `demographic` and `navigation`). As anticipated above, this structure is a semi-lattice since



```
<!-- "user" Data Schema (excerpt) -->
<DATA-DEF name="user.name" short-description="User's Name"
   structref="#personname">
    ...
</DATA-DEF>

<DATA-DEF name="user.home-info" short-description="User's Home Contact
   Information" structref="#contact">
    ...
</DATA-DEF>

<!-- "contact" Data Structure (excerpt) -->
<DATA-STRUCT name="contact.postal" short-description="Postal Address
   Information" structref="#postal">
</DATA-STRUCT>

<!-- "postal" Data Structure (excerpt) -->
<DATA-STRUCT name="postal.name" structref="#personname">
</DATA-STRUCT>

<!-- "personname" Data Structure (excerpt) -->
<DATA-STRUCT name="personname.given" short-description="Given Name">
    ...
</DATA-STRUCT>

<DATA-STRUCT name="personname.family" short-description="Family Name">
    ...
</DATA-STRUCT>
```

(a)

Figure 1. A small section of the P3P base data schema: (a) XML description, (b) graphical representation.

individual items may belong to more than one element (e.g., both the paths `User.name` and `User.home-info.postal.name` refer to the same information item). In turn, categories classify data elements orthogonally with respect to both data structures and elements, thus introducing in our model a new relation, different from the one introduced so far, that we call *member-of*. The overall structure is particularly useful at enforcement time, that is, when deciding whether a disclosure policy can be applied to a piece of personal data or not. For instance, a privacy preference could apply either to all occurrences of the `given` data element of the `Personname` structure or just to the specific context `User.home-info.postal.name.given`. Precedence and combining criteria must therefore be defined for all the three possible relations of our model (*is-a*, *part-of*, and *member-of*). In our approach, the informational content of users' credentials is modeled in a similar way, taking advantage of *is-a* sub-typing to represent variations of a base credential. For instance, legislation of different countries may require different data elements to appear within the same credential; however concrete definitions can be brought under the same umbrella by linking them to an abstract concept via *is-a* sub-typing.

3. Requirements for P3P extension

Current P3P architecture lacks the necessary expressiveness to represent all requirements of digital identity management. For instance, P3P was not designed to provide any degree of choice between the different credentials a user could provide to be granted access to a resource or service. However, our model can be used to guide the extension of P3P data schema to provide the required expressive power. Specifically, we aim at integrating *declarations* (i.e., uncertified data provided by the user itself) and *credentials* (i.e., certificates signed by third parties) in the same context and specify preferences over them so that transactions can be carried out with the least recourse to the user intervention and the least disclosure of data [2].

Declarations represent personal data provided by the end user and stored by the digital identity management system for later use. Personal data values are then mapped to a tree of P3P data elements. This tree is obtained starting from the P3P data schema's semi-lattice of dependencies and replicating elements that have multiple ancestors so that each replica has a single ancestor. Replication is necessary because the user could provide different values for the same data element according to different contexts. For instance, each mem-

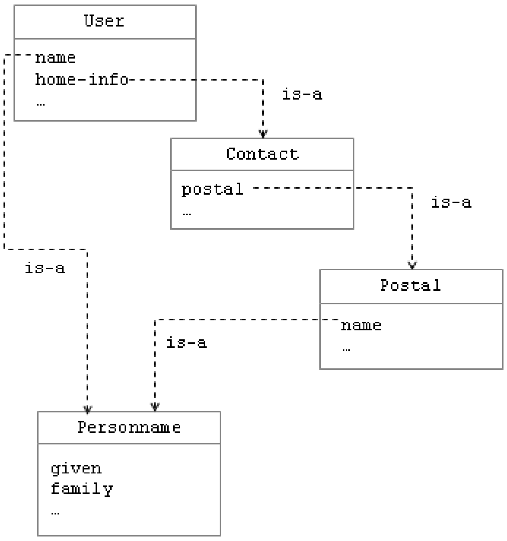


Figure 2. Class view of the same section of the P3P base data schema.

ber of a family could provide the same value for `User.credit-card-info.name` when booking a flight, while obviously providing distinct values for `User.name`. The redundancy of data elements in different data structures can also be used for suggesting options when a value of the former is missing in the specific data structure.[1] Credentials provided by certification authorities can be downloaded to the local system or just referenced by the management system when requested by a service provider. However, the inner structure of the credential should be provided by the certification authority so that it can be mapped to policies in the local system. Note that no assumption is made on the actual format of the credential, whether it is provided as a whole or it is possible to enucleate single elements (e.g., the `date-of-birth` out of a birth record).[2]

Figure 3 depicts a fragment of a sample `Portfolio`, an entity enclosing all the sensitive data stored by the system. For the sake of simplicity, here classes correspond to potential definitions of custom P3P data structures while attributes correspond to data elements, regardless of whether they point to a data structure or to a ground data type. Of course a custom data schema including custom data elements needs to be defined for this portfolio; however some of its elements (e.g., `name` in `CreditCardInfo`) correspond to elements of the base data schema of Fig. 1. We use uppercase names for actual instances of the defined data items, such as the `SWAP04` attendance certificate, and also thin boxes. When not labeled, relations are of type *subclass-of*. We introduce this relation to avoid defining redundant attributes and to provide visual cues of dependencies. With reference to Figure 3, any policy rule specified on structure `CreditCardInfo` will also apply to all of its descendants.

Figure 3 shows different kinds of entities:

1. Built-in entities expressing the system's functional requirements, such as class `Profile` allowing to store information according to a given user profile so that multiple users can share the same data. For instance, a single credit card could be shared by a whole family, possibly with constraints on the amount that can be charged.

2. Entities describing the inner structure of credentials and grouping declarations into classes according to a shared ontology built from various sources. For instance, class `CreditCardInfo` represents the standard information associated with a credit card.

3. Entities representing the composition of atomic data items and more general classes into higher-level abstractions, giving the user a way to categorize data in a custom fashion. For instance, class `AttendanceCertificate` is created to arbitrarily group a set of `AccreditationCertificate`s.

4. Entities embodying instances of concepts such as the actual values representing the user's `VISA` credit card.

[1] Different preferences could also be assigned to the `name` data element within the limits of the single context, perhaps depending upon whether or not the value is the same, or the preferences could be set to span across distinct instances of the same data element.

[2] Furthermore, zero-knowledge proof technologies such as the Idemix [8, 3] credential system could reduce the need for the actual release of data.

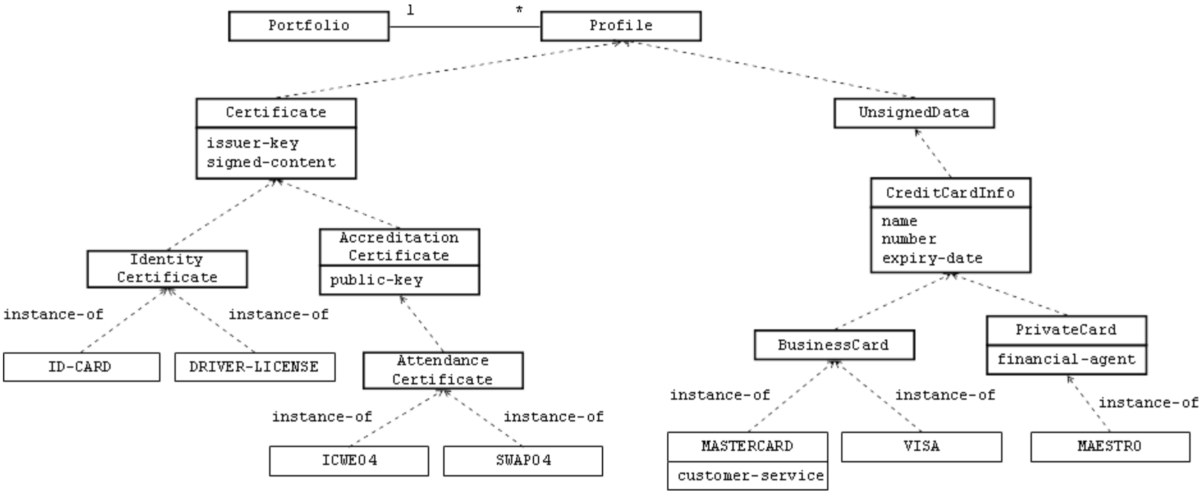


Figure 3. A sample portfolio.

In the example of Figure 3, it is clear that the personal data owner intervention was not limited to grouping data items or classes enveloping them with custom concepts such as `AttendanceCertificate`. She has also enriched the normative definition of class `CreditCardInfo` by sub-classing it with classes `BusinessCard` and `PrivateCard` and then extending the latter with the user-defined attribute `financial-agent`. Finally, she added to the single instance `MASTERCARD` the user defined attribute `customer-service`, pointing to a set of contact information. Correspondingly, policy definition languages should allow this kind of flexibility in the definition of data items to be requested or protected, together with the crosscutting classification of data items into categories.

4. Using Semantic Web languages for Representing Heterogeneous Personal Information

After defining the internal structure of credentials so that the gathering of fine-grained personal data can be carried out with no ambiguity, it is necessary to formulate the procedure enforcement uses to check the actual content provided by the negotiating parties against some underlying formal definition. A first step toward enabling these distinctions would be to define a credential taxonomy using XML Schema [15]. The XML Schema language allows one to map the associated information with structured data types, thus enabling further refinement and automatic evaluation. For instance, starting from an XML schema representing a driver license of a given country, it is straightforward to tailor a credential template for validating drivers of a specific city by restricting the values allowed to the corresponding element of the former. Such a definition can then be used to verify the credentials provided by a user as instance documents with the standard and widely implemented technique of XML validation.

On the other hand, the XML Schema language does not allow the specification of complex relationships such as those binding data type definitions to their roles inside the user's portfolio or the policies of a service provider.[3] Moreover, run-time data values are generally not known in advance (e.g., the current date at the time of transaction). For instance, a service provider willing to check whether or not a user's age is above 18 and is given the user's ID card should

[3] Much in the same way, the XACML policy definition language used in [4] needed to be extended in order to become semantics-aware.

be able to compute her age with respect to the current date. While it is possible to express this condition by combining both these values to obtain a `duration` data type, such a transformation can only take place in the instance documents and cannot be performed at the schema level. A wrapping mechanism should then be conceived to extend the semantics of XML Schema for representing application-level conditions. Our model of the P3P data schema can be used as a base ontology for expressing the meaning of data contained in the user's portfolio. Furthermore, it is possible to enrich our ontology with a specific task ontology representing a credential as a whole, relating it to the enclosed data elements, and adding facilities for the diachronic evolution of its normative definition (e.g., the migration process of banking records due to the Euro's introduction). Using the OWL syntax [14] (e.g., as shown in the W3C Note [7]) it is possible to take advantage of the reasoning features associated with the language. In this case, it would be also necessary to map the privacy preference language to the OWL syntax so that policies and requirements associated with them can be exchanged as triples.

4.1. Representing Privacy Preferences

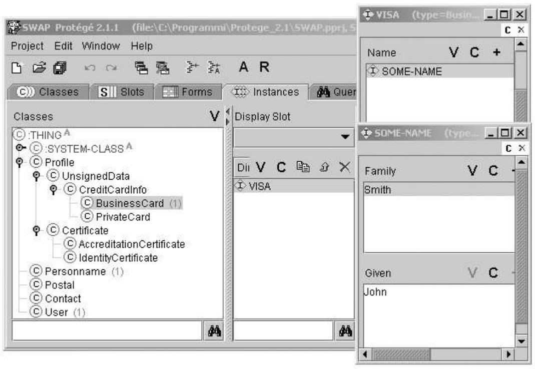
We envision three different techniques for taking advantage of metadata encoding personal identity information's structure and meaning for expressing privacy preference policies:

1. In the *offline* approach personal identity data can be used as a guideline for tailoring the access control infrastructure at policy writing time: changes in the ontology underlying personal data can be notified to end users and system administrators but have no impact on the policies defined. In this case a versioning system is required to check whether client and server are both referring to, other than the same credential, also the same version.
2. In the *online* approach, personal identity information's ontology is used at policy evaluation time so that the whole knowledge base can be updated according to the life cycle of its components. Changes in the ontology will be notified so that policies and user data locally stored by servers are kept consistent.
3. Finally, the *inverse* approach validates existing policies according to the metadata contained in the ontologies. Changes in the ontologies are propagated as in the previous case, but end users and system administrators are given hints to spot inconsistencies in the defined policies (e.g., notifying that different preferences are applied to distinct instances of the same data element).

To clarify the differences between these three approaches, consider some changes occurring to the definition of the `CreditCardInfo` data structure of Figure 3.

- In the offline approach the system administrator and the end user could be notified that the structure of a data item referenced in Access Control and Privacy rules respectively have been modified: the human agent at both sides can then choose whether to upgrade to the new version, revising the rules affected by the changes. This can also happen in the middle of a transaction as soon as a version mismatch in the data items being negotiated is reported.
- In the online approach the formal definitions of declarations and credentials are kept consistent with local copies and the administrator is fully aware of the life cycle of each component and can revise the rules accordingly. As in any other approach the system should provide default rules for handling `CreditCardInfo` data items until the rules affected by the changes are not checked out.
- In the inverse approach the synchronization mechanism is the same as in the online approach. The difference is that, other than being notified the changes as they occur, the administrator could be notified that just two instances of credit card out of three share a rule, requiring for instance a SSL connection for any transaction to take place: probably the rule should be applied to the `CreditCardInfo` data structure directly, thus affecting all its children.

In both the online and the inverse approaches, the distributed design of the knowledge base can lead to the exponential growth of the effort needed to keep it up to date. Therefore it is necessary to give up the



Name: SOME-NAME

Family: Smith

Given: John

Figure 4. The ontology integrating the entities of Figure 3 with P3P data structures.

pure “pull” model of current Web access and investigate the possibility of implementing “push” mechanisms. Figure 4 is a screenshot from Protégé [13] showing a OWL DL ontology integrating the structures of Figure 3 with the P3P data schema elements introduced in the paper: both the `BusinessCard` and `User` data structures have instances referencing the same instance of the `Personname` data structure `SOME-NAME`. “John” and “Smith” are values associated with the `given` and `family` properties.

5. Conclusions and future work

This paper has outlined a structural model of P3P data schema semantics. Encoding of our model in terms of Semantic Web languages like OWL have also been discussed. The reasoning capabilities of OWL, in contrast with the unconstrained representation capabilities of RDFS, allows the automatic deduction of information implied, whereas not explicitly stated, in the knowledge base. Different degrees of expressiveness available in the OWL language can bound the complexity of the reasoning process to meet execution constraints. However, much work is still to be done before this encoding can be used in practice. For instance, current OWL reasoners are only required to support the `xsd:integer` and `xsd:string` datatypes, while our model requires the full expressiveness of XML Schema in the definition of data type properties representing the portfolio items. We also need to constrain values allowed by such properties so that it is possible to specify, among the possible instances of a given credential, those satisfying the requirements imposed by a policy. We plan to address this issue in a future paper.

Acknowledgments

This work was supported in part by the European Union within the PRIME Project in the FP6/IST Programme under contract IST-2002-507591 and by the Italian MIUR within the KIWI and MAPS projects.

References

[1] *A P3P Preference Exchange Language* (APPEL) - World Wide Web Consourtium - http://www.w3.org/TR/P3P-preferences/

[2] P. A. Bonatti, P. Samarati - *A Uniform Framework for Regulating Service Access and Information Release on the Web* - Journal of Computer Security 10(3): 241-272 (2002)

[3] J. Camenisch, and E. Van Herreweghen - *Design and Implementation of the idemix Anonymous Credential System* - IBM Zurich Research Laboratory - http://www.zurich.ibm.com/ jca/papers/camvan02.pdf

[4] E. Damiani, S. De Capitani di Vimercati, C. Fugazza, and P. Samarati - *Extending Policy Languages to the Semantic Web* - International Conference on Web Engineering 2004: 330-343 http://www.icwe2004.org/

[5] E. Damiani, S. De Capitani di Vimercati, P. Samarati - *Managing Multiple and Dependable Identities* - IEEE Internet Computing 7(6): 29-37 (2003)

[6] *eXtensible Access Control Markup Language* (XACML) - Organization for the Advancement of Structured Information Standards - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[7] G. Hogben - *P3P Using the Semantic Web* - W3C Note 16th August 2004

[8] *Idemix anonymous credential system* - IBM Zurich Research Laboratory - http://www.zurich.ibm.com/security/idemix/

[9] *Platform for Privacy Preferences* (P3P) - World Wide Web Consourtium - http://w3.org/P3P/

[10] *Privacy and Identity Management for Europe* (PRIME) - European RTD Integrated Project - http://www.prime-project.eu.org/

[11] *Public Key Infrastructure* (PKI) - National Institute of Standards and Technology - http://csrc.nist.gov/pki/

[12] *RDF Vocabulary Description Language* (RDFS) - World Wide Web Consourtium - http://www.w3.org/TR/rdf-schema/

[13] *The Protégé project* - Stanford University - http://protege.stanford.edu/

[14] *Web Ontology Language* (OWL) - World Wide Web Consourtium - http://w3.org/2004/OWL/

[15] *XML Schema* - World Wide Web Consourtium - http://w3.org/XML/Schema

[16] Ting Yu, Ninghui Li, Anne Anton - *A Formal Semantics for P3P* - ACM Workshop on Secure Web Services, Oct. 2004