Enforcing P3P Policies Using a Digital Rights Management System

Farzad Salim, Nicholas Paul Sheppard and Rei Safavi-Naini

[1] School of Computer Science and Software Engineering,
University of Wollongong, NSW 2522, Australia
fsalim,nps@uow.edu.au
[2] Department of Computer Science, University of Calgary,
2500 University Drive, NW, Calgary T2N IN4, Canada
rei@cpsc.ucalgary.ca

Abstract. The protection of privacy has gained considerable attention recently. In response to this, new privacy protection systems are being introduced. SITDRM is one such system that protects private data through the enforcement of licenses provided by consumers. Prior to supplying data, data owners are expected to construct a detailed license for the potential data users. A license specifies whom, under what conditions, may have what type of access to the protected data.
The specification of a license by a data owner binds the enterprise data handling to the consumer's privacy preferences. However, licenses are very detailed, may reveal the internal structure of the enterprise and need to be kept synchronous with the enterprise privacy policy. To deal with this, we employ the Platform for Privacy Preferences Language (P3P) to communicate enterprise privacy policies to consumers and enable them to easily construct data licenses. A P3P policy is more abstract than a license, allows data owners to specify the purposes for which data are being collected and directly reflects the privacy policy of an enterprise.

1 Introduction

Information privacy is regarded as the right of individuals to determine for themselves when, how, and to what extent information about them is communicated to others. The concern about information privacy is growing for consumers who may need to release their personal data to enterprises in exchange for a service. In response to this concern, enterprises publish a *privacy policy* that is a representation of different legal regulations, promises made to data owners, as well as more restrictive internal practices of the enterprise.

Traditionally, privacy policies were written in natural languages. However, informal privacy policies inherit the potential ambiguity and mis-interpretation of natural text [16]. This raises two problems, first, such policies are difficult for consumers to read and understand, and second, controlling the enterprise data practices using such policies is impractical.

To address the first problem, the World Wide Web Consortium has proposed a standard policy language, the Platform for Privacy Preferences (P3P), to enable enterprises to construct machine-readable privacy policies [6]. P3P policies can be read, summarized and matched against users' privacy preferences by P3P-enabled browser software (*P3P agents*). Therefore, data owners can be prompted on exactly what data is collected, for what purposes this data is to be used and how long it is retained. Background information about the P3P language is provided in Section 2.4.

Although enterprises who have posted P3P policies promise specific data usage, they still require internal mechanisms to enforce those promises. In other words, publishing a P3P policy does not provide any technical guarantee that enterprises act according to their policies once they have obtained user's personal data. To address this problem, privacy protection systems such as *E-P3P* [10], *Tivoli* [3] and *SITDRM* [3] [15] are emerging.

Tivoli, being the commercial version of *E-P3P*, is a privacy protection framework that extends traditional access control systems by adapting a privacy oriented language known as *EPAL* [12]. The language provides a syntax that allows a *policy auditor*[4] to specify privacy rules. In addition, *EPAL* has operational semantics that govern the interpretation of the rules with respect to an access request. Hence, an authorization decision can be made when a data user requests to access a private data.

SITDRM uses another approach to the privacy enforcement problem. It adopts the extended Digital Rights Management (DRM) model that was proposed by Korba et al. [11] and implemented by Sheppard et al. [15]. The core concept in SITDRM is the use of licenses that are formulated by consumers and enforced by a digital rights management system. A license is a digital data file that specifies usage rules for the collected data. A rule may specify a range of criteria, such as the person to whom the right is issued, the frequency of access, license expiry date, restriction of transfer to other devices, etc. Hence, such licenses can express the notions of privacy policies (e.g., obligations or conditions, etc.) under which the data must be used, or the type of actions that can be performed on the collected data. We will give an outline of the relevant components of SITDRM in Section 2.3.

Whilst the SITDRM approach binds the enterprise data handling to the privacy promises made to customers, it has some limitations that we would like to address in this work. First, data subjects in SITDRM are obliged to construct an MPEG REL license. However, this task cannot be handled by an average customer because MPEG REL has a complex syntax and semantics and was designed to be used by policy auditors for specifying concrete access control rules.

Further, to create an MPEG REL license, the consumer must have knowledge of the identity of the user (or role) that the license is to be issued to. SITDRM

[3] Smart Internet Technology Digital Rights Management

[4] The policy auditor is a person responsible for writing enterprise privacy policies. In the legal context, this person is referred to as the Chief Privacy Officer (CPO).

currently assumes consumers are provided with such information. However, for many real world scenarios this is not practical as such knowledge about the roles/employees may reveal the internal structure and data flow of the enterprise. For example, a bank customer would be able to know who in the bank has access to customer's account balances.

In addition, SITDRM needs to handle the dynamicity of the organization's structure. The roles/users within an enterprise change more frequently than the purposes for which the data is being collected. Currently, customers are obliged to provide a new license each time the roles/users (license holders) change.

Finally, SITDRM requires a systematic approach for creating templates for collecting privacy preferences. Currently it assumes that there exists a human user (privacy officer) who is aware of the enterprise privacy policy and can construct license templates. These templates are then used by customers to create a concrete MPEG REL license. However, in an enterprise which may collect data at more than one point with different privacy rules the maintenance and synchronization of the policies with these templates becomes impractical.

To address the above, we extend SITDRM by employing P3P to allow consumers to modify a subset of an enterprise P3P policy for expressing their privacy preferences. P3P preferences are more abstract than a license and allow data owners to specify the purposes for which data is to be collected. Hence, they do not need to know concrete roles, rights or access conditions. Further, P3P preferences reveal less information regarding the enterprise's internal structure and eliminate the need for their re-issuing, when an internal role changes and the purpose remains the same.

Despite these advantages, P3P preferences are not directly enforceable, so they need to be transformed into MPEG REL licenses that can be enforced by SITDRM. In this paper we outline the difficulty of such as a translation and propose a practical approach for mapping a P3P statement to an MPEG REL grant.

We have also extended SITDRM's design and implementation by adding two new components, the *P3P Agent* and the *Mapping Console*. The P3P agent provides a systematic approach for collecting an organization's P3P policy and constructing a P3P preference template for data owner's to customize and express their privacy preferences. The mapping console assists CPOs in specifying the mapping rules for constructing MPEG REL licenses.

The rest of this paper is organized as follows: In Section 2 we will provide the necessary background. In Section 3 we will show the architecture of our new P3P-Enabled SITDRM. Section 4 will describe the necessary mapping rules for transforming P3P preferences into an MPEG REL license. Section 5 will discuss how we can systematically automate a preference form from an enterprise's (P3P) privacy policy. We conclude the paper with an a discussion of outstanding issues and conclusions in Sections 6 and 7.

2 Preliminaries

This section will briefly describe the Digital Right Managements (DRM) model for data protection, some components of SITDRM, MPEG REL and the P3P language. Interested readers may refer to [15, 6, 2] for further details.

2.1 DRM

Digital rights management provides protection for information by making access to information depend on satisfying the conditions imposed by a *license* written in a machine-enforceable *rights expression language*. DRM technology is widely used in copyright protection applications, but can also be applied to privacy protection [11, 15] by developing licenses that represent individual's preferences for use of their personal information.

2.2 MPEG-21

The MPEG-21 Framework [2] is a framework for creating, distributing, using and controlling multimedia content currently under development by the Motion Picture Experts Group (MPEG). Of particular interest to us are three components in the MPEG-21 framework: *Digital Items (DI)*, *Intellectual Property Management and Protection (IPMP)* and the *Rights Expression Language (REL)*.

Digital Items The core notion in MPEG-21 is a digital item [2], which represents a collection of multimedia objects. Digital items are described using the XML-based digital item declaration language (DIDL), which organizes content and meta-data. For the purposes of this paper we consider digital items to be the encapsulation of private data that needs to be protected.

Intellectual Property Management and Protection Intellectual Property Management and Protection is MPEG's term for digital rights management [2]. MPEG-21 does not define a digital rights management system, but assumes that IPMP functionality is provided by vendor-specific IPMP tools that can be downloaded and made accessible to the terminal as necessary. IPMP tools may implement basic functions such as decryption and watermarking, or may implement complete digital rights management systems in their own right.

Rights Expression Language Though MPEG-21 does not define a full digital rights management system, it does define a rights expression language known as MPEG REL [2] for creating machine-readable licenses. An MPEG REL license is structured as a collection of *grants* issued by some *license issuer*. Each grant awards some *right* over some specified *resource* to some specified *principal*, that is, user of a resource. Each grant may be subject to a *condition*, such that the right contained in the grant cannot be exercised unless the condition is satisfied.

In order to perform some action on a resource, a user (principal) must possess a license containing a grant that awards the right to perform that action on that resource, and satisfy the associated condition. This must be checked by the terminal prior to exercising the right.

MPEG REL is defined as a collection of three XML schemas, called the core schema (denoted by the XML namespace prefix r in this paper), the standard extension schema (prefix sx) and the multimedia extension schema (prefix mx). In addition, the authors in [15] introduced a *privacy extention schema* with prefix px for use in the proposed privacy protection system (SITDRM). Figure 3 shows an example of an MPEG REL grant allowing a principal (r:keyHolder) identified by his/her public key to print a resource (mx:diReference) identified by a digital item identifier (smartinternet:doc1). The principal is only permitted to print the resource once (sx:ExerciseLimit).

```
<r:grant>
  <r:keyHolder>
     <dsig:KeyValue> ...  </dsig:KeyValue>
  </r:keyHolder>
  <mx:print/>
  <mx:diReference>
    <mx:identifier> smartinternet:doc1 </mx:identifier>
  </mx:diReference>
  <sx:ExerciseLimit>
    <sx:count> 1 </sx:count>
  </sx:ExerciseLimit>
<r:grant>
```

Fig. 1. A License

2.3 SITDRM

SITDRM is an implementation of MPEG-21 IPMP for privacy protection. It provides a framework within which content providers can control the use and distribution of personal data (content) through the enforcement of data licenses. In SITDRM each resource that is protected by an IPMP tool is referred to as a *governed* resource. Each governed resource is associated with a plain text identifier and an IPMP information descriptor that associates the resource with a license and describes the IPMP tools required to access the resource. If the conditions of the license are satisfied, the terminal must obtain and instantiate the IPMP tools in order to access the resource.

Figure 2 shows an overview of the SITDRM system, where a data controller (e.g., bank) requires information to be collected from data subjects (e.g., customers). All of this information is stored in some central database. In addition,

there are some data users (e.g., employees) that require access to the information in order to carry out their jobs and provide service to the customers.

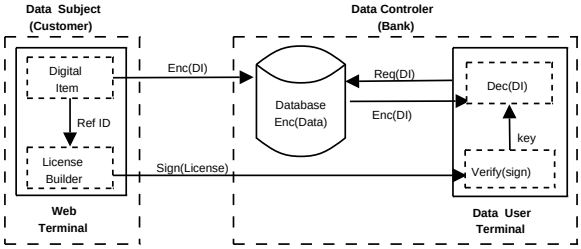

Fig. 2. SITDRM Architecture

Customers submit their information via a form on the bank's web site. For example, a document containing the customer's credit card number and postal address are formatted as an XML document. At the same time, customers design an MPEG REL license that describes how this information may be used.

Upon submitting the form, the customer's web browser converts the resulting XML document into the governed resource of an MPEG-21 digital item, and issues a license designed by the customer. The governed item and issued license are then transmitted to the data controller for storage.

Employees who require access to a customer's data may download the governed item from the data controller. Upon attempting to perform some action on the item, the employee's terminal asks the data controller for a license that authorizes this action. If an appropriate license is found, the action is permitted to continue. Otherwise, the action is rejected.

2.4 P3P

P3P [6] is a standard developed by the W3 Consortium for assisting web users to discover and evaluate the privacy policies of on-line service providers. P3P consists of an XML-based *language* for expressing the privacy policies of service providers, and a *protocol* for associating a P3P policy to collected data and locating the privacy policy file relevant to any particular data collection action. A P3P policy file is composed of a sequence of *statements*, each containing five elements described below. Figure 3 shows a typical statement in a P3P policy.

- a *purpose* for which this data will be used;
- a *recipient* to whom this data may be communicated;
- a *retention* policy according to which the data will be discarded;
- a *data group* of the data to which this statement applies; and
- a *consequence*, being an informal reasoning behind the collection of this data.

```
<STATEMENT>
  <CONSEQUENCE>
    We will access your credit card records to process your loan
    requests. We have the right to retain these information
    for a year.
  </CONSEQUENCE>
  <DATA-GROUP>
    <Credit-Card-History required="always"/>
    <Credit-Card-Number  required="opt-out"/>
  </DATA-GROUP>
  <PURPOSE>
     <loan required="opt-out">Loan and Finance</loan>
  </PURPOSE>
  <RETENTION>  <one-year />  </RETENTION>
  <RECIPIENT>  <ours />     </RECIPIENT>
</STATEMENT>
```

Fig. 3. A P3P Statement

Despite many advantages of the P3P language such as providing a standard way of communicating privacy policies to users and allowing for the automation of matching privacy preferences and privacy policies [6, 1, 9], the language is subject to some criticisms. First, P3P policies are subject to multiple interpretations and different software agents that read them may arrive at different conclusions for the same policy [17]. Second, privacy polices are offered on a take-it-or-leave-it basis by the service provider, and may not reflect the actual desires of users [4]. Third, the policies given in a P3P policy are not automatically enforced [5], requiring dissatisfied data subjects to resort to legal action.

Yu, et al. [17], address the first problem by proposing relational formal semantics for P3P and introducing some integrity constraints. Namely they introduce two types of semantics by which P3P statements can be interpreted, *data-centric* and *purpose-centric* semantics. The term semantics in this context refers to the relationships among the four major components (purpose, recipient, retention and data) of a P3P statement.

In purpose-centric semantics, a data item along with a purpose determines other elements (i.e., recipients and retention) in a P3P statement. So, each statement takes only one purpose and other elements in the statement are centered around that purpose. The rationale behind the purpose-centric semantics is that certain data is sometimes used for multiple purposes, depending on the specific purposes, and the data may be kept for different periods of time. Hence, binding the data and purpose of each statement avoids potential inconsistencies in a statement. For the rest of this paper we assume that our P3P policies have purpose-centric semantics.

3 P3P-Enabled SITDRM Architecture

In order to address the aforementioned limitations of SITDRM, we employed P3P in conjunction with MPEG REL licenses. The P3P protocol is adopted as a standard approach for the communication of privacy policies during data collection [5].

Further, at one end the P3P language is used for presenting privacy policies to customers and collecting their privacy preferences, and at another end, MPEG REL is employed for specifying enforceable licenses (with concrete access control rules). The combined use of these two languages bridge the gap between the abstraction required for consumers to specify their privacy preferences and the precision needed for a license to be enforceable by user's terminals.

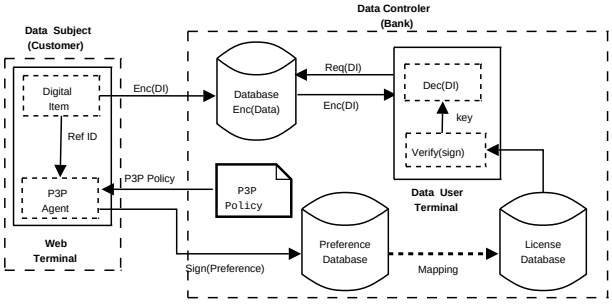

Fig. 4. P3P-Enabled SITDRM Architecture

As shown in Figure 4 the new architecture for SITDRM introduces the following new components: a *P3P policy*, a *P3P agent* and *mapping rules*. We assume that there is a Chief Privacy Officer *(CPO)* who writes a P3P policy that specifies the data handling of the enterprise. The P3P policy can then be retrieved and represented to a policy subject by using the P3P agent that is embedded within the data subject's web terminal. In addition, the P3P agent allows the data subject to modify the enterprise policy such that it matches his/her *privacy preferences*. Privacy preferences are then digitally signed and transferred to the enterprise where they are stored. These signed privacy preferences represent the consent of the data subjects. Since the P3P preferences are abstract, they cannot be directly enforced by the data user's terminals, hence, they need to be transformed into their MPEG REL license(s).

In the following section we will explain how a P3P policy can by converted to an MPEG REL license enforceable by SITDRM. We say an MPEG REL license

[5] Since we simply adopt the P3P protocol, this paper does not elaborate on the policy communication aspect. Interested readers may refer to P3P specification for more details.

corresponds to a P3P preference if the license is constructed using the mapping methodology that we will introduce in Section 4.

4 Constructing a License from a P3P preferences

In order to convert P3P preferences to an MPEG REL license we will identify the correspondence between the elements in both languages and provide a set of mapping rules that take a P3P policy as an input and return an MPEG REL license(s). However, our aim is not to specify a new standard universal vocabulary for P3P or MPEG REL. Because a particular mapping of P3P elements to MPEG REL elements is enterprise dependent (e.g., role names vary), having a global vocabulary or a generic mapping would be impractical. Rather, we would like to introduce a practical approach to transform a subset of P3P policies into MPEG REL licenses. Figure 5 illustrates the associations that we would like to introduce between the components of a P3P *statement* and an MPEG REL *grant*. Those connections that are represented with dottedlines show the areas where there is no direct relationship between the two components.

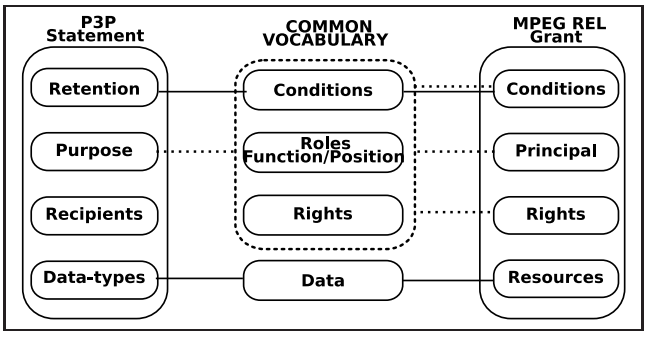

Fig. 5. P3P statement & MPEG-REL grant

In the following sections we will discuss a methodology that permits the conversion of P3P preferences into one or more MPEG REL licenses. We have incorporated the mapping into a tool, the *Mapping Console*, that assists CPOs by providing the following. First, a central point where the vocabulary necessary (e.g., data elements) for creating an enterprise P3P policy and an MPEG REL license is introduced and stored. This is particularly of interest as the syntax and semantics behind the vocabularies used in P3P and MPEG REL determines the correctness of the mapping rules that are being introduced. Second, like any access control management application, the Mapping Console allows the CPO to specify the *roles*, *principal* and *principal/role* assignments through the role specification window shown in Figure 6. Third, given this contextual information, it allows the CPO to customize the mapping rules as well as automating the process of constructing MPEG REL licenses from P3P preferences.

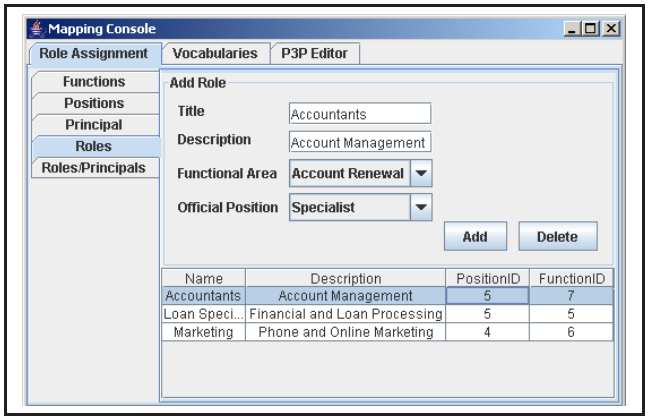

Fig. 6. The Mapping Console: Role Assignment Tab

4.1 Purpose

The most conspicuous difference between MPEG REL and P3P language is the latter's use of `purposes`. In the MPEG REL model, a `purpose` must be interpreted as some combination of a particular `principal` exercising a particular `right` under certain `conditions`. In what follows we will describe our approach for determining `principal(s)`, necessary `rights` and `conditions` for a license such that it complies with the `purpose` of the corresponding P3P preference.

Principals: In order to realize the correlation between a `purpose` (in P3P) and a `principal` (in MPEG REL) we need to consider the relationships between data users in an enterprise and the tasks that they perform. The definition of such a relation is directly dependent upon the access control model adopted. Here, we consider access control systems that are based on the Role Based Access Control (RBAC) model [13]. In this model several roles are identified, and associated with them are some rights and conditions, and principals are assigned to roles which enable them to perform certain tasks on data that is predefined for that role.

We adopt the model proposed by Schaad et al. [14], where roles in enterprises are composed of a description of a *function* and an *position* within the organizational hierarchy. Functions represent the type of duties that a role is based on, such as loan processing, promotion & marketing, etc. Typical positions could be that of the ordinary clerk or group manager. An example of a role would be loan processing/group manager, indicating that the principal of the role performs a loan processing function and holds the official position of a group manager. The following table shows some typical roles in a bank.

In this model, the job functions indicate the purposes for which a role may use the data. Hence, this indirect relationship between a purpose and a principal (through roles) allows us to determine those data users involved in carrying out a purpose. Since a role is a composition of a function and a position, several

Role ID	Function	Position
A	Promotion & Marketing	Manager
B	Promotion & Marketing	Officer
C	Loan processing	Officer
D	Finance	Specialist
E	Account Management	Bank Manager

Table 1. Roles

roles may be involved in carrying out a purpose. For example, a marketing officer, marketing manager and delivery person may all work under the function "Promotion & Marketing". Figure 7 shows the relationships between purpose, roles, access rights and principals in the system.

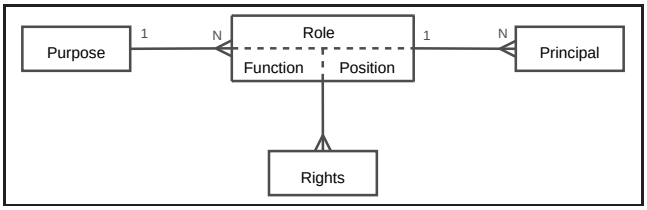

Fig. 7. Role Rights Relationship

Rights: The notion of *right* and *condition* in the P3P language are implicit within the element `purpose`. For example, two P3P purposes email-marketing and telemarketing may be instances of the same right, contact, but imply different conditions.

In a typical RBAC model, there exists a security officer who determines the appropriate rights for each role. We follow the same approach and assume that the rights for each role are determined by the CPO. Hence, given a P3P `purpose` we can identify (i.e., through roles) the `principal` and the `rights` for the corresponding license.

Conditions: Given a role, there may exist several *conditions* that must be satisfied before a member of the role can carry out an action. We have categorized the conditions into two classes. First, the conditions that are indirectly specified by the data subjects through the purpose for which their data can be used. Such conditions are implicit to some of the P3P purpose element, such as `contact` and `telemarketing`[6] . The first one only asserts that the customer may be contacted

[6] Note that due to the shortage of the space we skip providing the definition of the P3P elements. Interested readers may refer to the P3P specification.

and allows for any mode of contact and the second purpose which stipulates the only possible method of contact is phone.

In order to reflect these restrictions in MPEG REL licenses, we use the `ContactMethods` condition. This element contains an arbitrary number of elements of type `ContactMethodUri`, each of which has a mandatory `definition` attribute specifying a URI defining a particular method of contact. A typical contact method might include e-mail, telephone (using voice-over-IP) or the short messaging service (SMS). If the `ContactMethods` child is absent, any available contact method may be used for making contact.

The second class of conditions are those that are imposed by a CPO. These are usually application dependent conditions, derived from the security/privacy policy of the enterprise. Hence, data subjects may not necessarily be informed of the existence of these conditions. An instance of such conditions may impose a constraint that limits the access of principals (with certain roles) to a specific applications within the enterprise. Further, with the advent of wireless devices, location-based access control is also gaining more attention, so the CPO may like to ensure that the data is only accessible within specific zones in the enterprise.

4.2 Retention

The P3P language introduces five elements to specify the retention policy: `no-retention`, `indefinitely`, `stated-purpose`, `legal-requirements` and `business-practices`

Whilst the first two elements specify a destruction timetable, the other three indicate that the retention period is dependent on other factors such as legislation. Since a license requires a concrete time constraint in order to be enforceable, we are only able to map the first two retention elements. In addition to the above standard P3P vocabularies, we have also introduced a sub-element for `retention` that is used in our SITDRM application and allows a CPO to express the exact retention timetable. These elements represent a year, month or day that the data can be retained, (e.g., `one-year` or `one-month`).

For those `retention` sub-elements that specify a time constraint such as `one-year` we use the `validityInterval` condition in MPEG REL that allows us to specify the duration for which the resource can be used. For example, let us assume a data subject formulates P3P preferences that specify that data may be used for one year. In order to construct an appropriate license condition, SITDRM adds the condition shown in Figure 8 to the corresponding license.

The `no-retention` element means "Information is not retained for more than a brief period of time necessary to make use of it during the course of a single online interaction". We consider the `no-retention` element in a P3P statement to mean that the collected data can only be used once (in a single interaction). Hence, in order to specify this as an MPEG REL condition we use the `ExerciseLimit` element which allows us to specify the number of times that the license can be used. A simplified version of such a condition was shown in Figure 1.

Trivially, when the P3P retention element is `indefinitely` we do not specify any time condition for the MPEG REL license.

```
<r:validityInterval>
  <!-- Time the record is submited -->
  <r:notBefore>2006-12-01T00:00:00</r:notBefore>
  <!-- Destruction time -->
  <r:notAfter>2007-12-01T00:00:00<r:notAfter>
</r:validityInterval>
```

Fig. 8. An MPEG REL condition: Time Interval

4.3 Data-Type

Both a statement in P3P and a grant in a license refer to a piece of information that needs to be protected. In MPEG REL the `resource` element contains a reference to the actual data that the license is about to provide access for. In P3P language the `data-types` consist of sub-elements that specify the type of data that is being collected.

The `data-types` element may include sub-elements at different levels of granularity. It can refer to both aggregate data (categories) as well as more concrete pieces of information, such as, name, e-mail address or credit card number. In P3P, personal data is classified into eighteen categories, including physical contact information, purchase information, demographic and socio-economic data, etc. For example, a person's gender, blood type, and date of birth belong to the category of demographic and socio-economic data. Thus, an organization may only specify the categories of personal data it wishes to collect rather than the concrete data elements. Although such an abstraction simplifies the task of specifying a P3P policy, it will adversely effect the granularity of the licenses. Therefore, in this work, we avoid using categories and only allow the specification of P3P policies with concrete data-types. Currently we use the `user-data` element in P3P base data schema [6], which consists of typical concrete data elements such as, name, birth-date, phone-number and postal-address, etc.

The only major difference between the `data-types` and the `resource` is that the latter contains reference to the particular data that is being collected. P3P `data-types`, on the other hand, do not refer to any particular record, they are only labels of what type of data the enterprise collects. In order to derive a license from a P3P preference we need to determine the reference to the collected data. The reference to the collected data can be trivially identified through the combined use of the unique record identification number that the data subject's web terminals assign to each transaction and the data types (e.g., phone, e-mail) specified in the P3P preferences.

4.4 Recipients

The reason for having the `recipient` element in P3P is to declare the third parties who may receive the collected data. The P3P language defines six possible recipients, `ours`, `delivery`, `same`, `other-recipient`, `unrelated`, `public`.

Although the recipient element may provide abstract information to data subjects regarding the sharing of their data, it does not play a significant role in mapping P3P preferences to an MPEG REL license. This is because the information that it provides can be extracted more precisely from purpose elements. For example, in a P3P statement where the recipient is `ours`, mapping the `purpose` allows us to determine the exact role(s) within the organization for which a license must be constructed.

Other than `ours`, the rest of the recipient classes suggest that the collected data is being shared with other parties. Hence, if there is a P3P statement in which `ours` is not part of its recipients we shall assume that we must not issue any license for the roles inside the enterprise, but only for those outside (within other organizations). However, since the recipient elements (e.g., `delivery, public`) are very abstract, we are unable to determine, whom (which role) within these third parties licenses must be issued to. One simplistic approach would be to assume that the role structure of both parties are identical, in which case, the roles, rights and conditions that were determined through purpose element can be used. But in reality, each organization has its own role model, so a more sophisticated approach must be taken. The current scenario of SITDRM assumes that the collected data is to be protected within the boundary of the enterprise. Hence, we can safely assume that only `ours` is used. Information sharing and cross-organizational privacy control using SITDRM will be our future work.

5 Specifying P3P Preferences

As described in Section 2, the intention of P3P is to provide a standard language to inform data owners of the global privacy practices within an enterprise. In reality, these policies, after being accepted by consumers, can also be considered as their privacy preferences. For example, consider a policy statement that states, the collected phone numbers will be used for marketing purposes. In this scenario, if a consumer (Alice) accepts this policy statement, we can safely consider the statement to be Alice's privacy preferences with respect to the use of her phone number. In the rest of this section we will describe how P3P policies can be used to collect data owner's preferences.

Recall that the elements that constitute a P3P statement may either be optional or compulsory. A policy statement that is composed of *optional* elements can be customized by data subjects to reflect their privacy preferences. For example, consider an arbitrary policy which indicates that Alice's credit card history is accessed to process her loan application and allows her to choose to be contacted via e-mail, telephone or fax. In contrast, those policy statements with *non-optional* elements indicate the areas where the operations that are performed on data are necessary for the enterprise and cannot be changed.

Based on the above concept, we have developed a *P3P Agent* that retrieves the enterprise P3P policy and constructs a template that enables the data subject to modify the P3P policy to create and submit his/her privacy preferences.

Fig. 9. P3P Preference Template

The idea of having client side agents that can retrieve P3P policy is not new. There are P3P agents such as AT&T Privacy Bird [7] that can parse the P3P policy and evaluate it against the consumer's privacy preference written in APPEL [8]. Unlike these tools our P3P agent introduces a notion of policy negotiation between consumers and the enterprise by allowing consumers to modify the P3P policy to construct their preference. However, users are not free to express any preference they wish: the scope of the negotiable policy is strictly defined by the enterprise and depends on what optional elements are included in the enterprise's P3P policy.

Our P3P agent collects the relevant P3P policy from the enterprise website and generates a *preference template*. A preference template is a graphical representation of the statements which constitute the enterprise P3P policy. The elements that constitute a statement in the template can either be fixed or modifiable, depending on their attributes (*always, opt-in and opt-out*). This allows the data owners to modify the policy based on their privacy preferences. We refer to the modified P3P policy as *P3P Preferences*. These P3P preferences will be digitally signed by customers (using their web terminal) and sent to the enterprise P3P preferences database. The pseudo-algorithms 1 and 2 in Appendix A show how a preference template is generated from a P3P policy.

Figure 9 shows a P3P preference template that was generated by using the P3P policy shown in Figure 3. As you can see for every statement in the policy, the informal description is followed by a set of modifiable check boxes for `data`, `retention` and `recipients`. These elements are centered around the `purpose` of the statement to highlight our purpose-centric semantics of the P3P language.

6 Future work

The implementation of the P3P agent can be extended by allowing data owners to specify their preferences in the form of APPEL rules. Hence, when the agent receives the P3P policy, it first evaluates the P3P rules with respect to the user's

APPEL rules and only shows the rules that do not match the current preferences. This will semi-automate the process of issuing P3P preferences.

Currently, we assume the elements (e.g., purposes) in a P3P statement are atomic elements and there are no hierarchical relationships between them. However, in reality purposes as well as data-types can be structured in a hierarchical form. One extension to the current P3P agent is to allow for the expression of these hierarchies. For example, to allow the data owner to see that the purpose "marketing" may indeed mean *direct marketing* and *online marketing* where direct marketing itself can consist of *phone marketing* or *fax marketing*.

Finally, as an extention to this work we investigate how organizations that implement SITDRM could share sensitive information, whilst ensuring the shared data are used with respect to the privacy policy of the organization that has collected the data.

7 Conclusion

In this paper we have discussed our approach for mapping P3P statements to MPEG REL grants, hence, to construct MPEG REL licenses from P3P preferences. Further, we have adopted this theory to extend the SITDRM architecture with a P3P handling component. Our extension will improve SITDRM and enable us to achieve the following goals. First, to facilitate the communication of P3P privacy policies with data subjects and enabling them to specify enforceable privacy preferences. Second, to provide a systematic way of creating templates, through which data subjects can specify their privacy preferences. Lastly, to ensure that the internal structure of the enterprise (users, roles) remains hidden from data subjects while they are formulating their privacy preferences.

References

1. A. Barth and J. C. Mitchell. Enterprise privacy promises and enforcement. In *WITS '05: Proceedings of the 2005 Workshop on Issues in the Theory of Security*, pages 58–66, New York, NY, USA, 2005. ACM Press.
2. J. Bormans and K. Hill. International standards organization. information technology - multimedia framework (MPEG-21) - part 5: Rights expression language. ISO/IEC 21000-5:2004.
3. A. Bucker, B. Haase, D. Moore, M. Keller, O. Koblinger, and H.-F. Wu. IBM tivoli privacy manager solution design and best practices. In *Redbooks*, 2002.
4. J. Catlett. Open letter to P3P developers and replies. In *In ACM Conference on Computers, Freedom and Privacy*, pages 157–164, 2000.
5. K. Coyle. P3P: Pretty poor privacy? a social analysis of the platform for privacy preferences (P3P).
6. L. Cranor, M. Langheinrich, M. Marchiori, and M. Presler-Marshall. The platform for privacy preferences 1.0 (P3P 1.0) specification. 2002.
7. L. F. Cranor, M. Arjula, and P. Guduru. Use of a P3P user agent by early adopters. In *WPES*, pages 1–10, 2002.

8. L. F. Cranor, M. Langheinrich, and M. Marchiori. A P3P preference exchange language 1.0 (APPEL 1.0). In *W3C Working Draft*, 2002.
9. G. Karjoth, M. Schunter, and E. V. Herreweghen. Translating privacy practices into privacy promises : How to promise what you can keep. In *POLICY '03: Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks*, page 135, Washington, DC, USA, 2003. IEEE Computer Society.
10. G. Karjoth, M. Schunter, and M. Waidner. Privacy-enabled services for enterprises. In *DEXA Workshops*, pages 483–487, 2002.
11. S. Kenny and L. Korba. Applying digital rights management systems to privacy rights management. *Computers & Security*, 21(7):648–664, 2002.
12. Research Report 3485, IBM Research. *Enterprise Privacy Authorization Language (EPAL)*, 2003.
13. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
14. A. Schaad, J. Moffett, and J. Jacob. The role-based access control system of a european bank: a case study and discussion. In *SACMAT '01: Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, pages 3–9, New York, NY, USA, 2001. ACM Press.
15. N. P. Sheppard and R. Safavi-Naini. Protecting privacy with the MPEG-21 IPMP framework. In *6th Workshop on Privacy Enhancing Technologies*, pages 152–171, 2006.
16. W. H. Stufflebeam, A. I. Antón, Q. He, and N. Jain. Specifying privacy policies with P3P and EPAL: lessons learned. In *WPES*, page 35, 2004.
17. T. Yu, N. Li, and A. I. Anton. A formal semantics for P3P. In *SWS '04: Proceedings of the 2004 Workshop on Secure Web Service*, pages 1–8, New York, NY, USA, 2004. ACM Press.

A Pseudo Codes to Construct a Preference Template

Algorithm 1 createPreferenceForm(policy)

```
1: while new(statement) do
2:    while new(purpose) do
3:       create a purpose block
4:       while new(data) do
5:          attribute = getAttribute(data)
6:          addOptions(data, attribute )
7:       end while
8:       while new(retention) do
9:          attribute = getAttribute(retention)
10:         addOptions(retention, attribute )
11:      end while
12:      while new(recipients) do
13:         attribute = getAttribute(recipients)
14:         addOptions(recipients, attribute )
15:      end while
16:   end while
17: end while
```

Algorithm 2 addOption(element, attribute)

```
1: if attribute = "always" then
2:    add(element, DISABLED)
3: else if attribute = "Opt-in" then
4:    add(element, UNCHECKED)
5: else if attribute = "Opt-out" then
6:    add(element, CHECKED)
7: end if
```

B Security Architecture of P3P-enabled SITDRM

There are three types of users that interact with SITDRM: a *data user*, who uses a trusted terminal to use the data, a *data owner* who provides the data and the *privacy officer*, who performs the tasks necessary for converting P3P preferences to licenses that can be used by data users.

In SITDRM, digital items are distributed in an encrypted form, and cannot be accessed without a secret key. Hence, the management of keys and licenses is of primary concern. In order to gain access to the content, a data user must obtain a valid license for the terminals that are being used. The license describes the terms and conditions under which the user may use the content, and also provides the user with the keys

required to access the content. Hence, terminals must be able to verify the authenticity and integrity of any license that grant rights over content. Every trusted terminal T has a private key $\bar{K}_T$ and its corresponding public key K_T. The private key $\bar{K}_T$ is known only to the terminal; in particular, it is not known to the human user of the terminal. This is to prevent users from accessing the data from any other terminal.

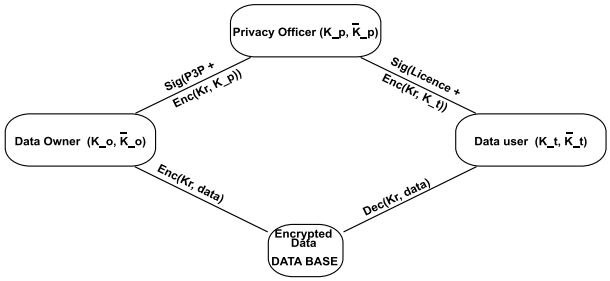

Fig. 10. P3P-Enabled SITDRM Security Architecture

The privacy officer has a private key $\bar{K}_P$ and corresponding public key K_P, and the authenticity of K_P can be verified by terminals using some public key infrastructure. The private key $\bar{K}_P$ is known only to the privacy officer.

The data owner has a private key $\bar{K}_O$ and the corresponding public key K_O. Similarly, the authenticity of K_O can be verified by the privacy officer using some public key infrastructure. In addition, the data owner has a master key $\mathcal{K}$ which is a symmetric key and known only to them. This key is used to encrypt individual resources as described in the following.

Every resource r in the system is associated with a public identifier i_r created by data controller to ensure the uniqeness of the records. All resources in the system are encrypted by a resource key K_r derived via a one-way function of the master key $\mathcal{K}$ and identifier i_r.

After generating K_r, the resources will be encrypted and stored in a resource database. Then the resource key K_r is encrypted with the public key of the privacy officer K_P and embedded to the P3P preferences of the data owner which were constructed using the P3P agent. The P3P preference is signed by the data owner before being sent to the privacy officer.

By having a P3P preference, the resource key and the mapping rules, the privacy officer is in a position to construct a license for data users. These licenses are only issued to the user's trusted terminals, so the user must supply the issuer with the public key K_T of the terminal T on which they wish to use the resource.

In addition to specifying the terms under which the resource r may be used, the license contains the resource key K_r encrypted by the terminal's public key K_T. Since the corresponding private key $\bar{K}_T$ is known only to the terminal T, only T is able to decrypt K_r and therefore decrypt the resource r.