Protecting Privacy with the MPEG-21 IPMP Framework

Nicholas Paul Sheppard and Reihaneh Safavi-Naini

School of Information Technology and Computer Science
The University of Wollongong, NSW, 2522, Australia
{nps,rei}@uow.edu.au

Abstract. A number of authors have observed a duality between privacy protection and copyright protection, and, in particular, observed how digital rights management technology may be used as the basis of a privacy protection system. In this paper, we describe our experiences in implementing a privacy protection system based on the *Intellectual Property Management and Protection* ("IPMP") components of the MPEG-21 Multimedia Framework. Our approach allows individuals to express their privacy preferences in a way enabling automatic enforcement by data users' computers. This required the design of an extension to the MPEG Rights Expression Language to cater for privacy applications, and the development of software that allowed individuals' information and privacy preferences to be securely collected, stored and interpreted.

1 Introduction

The increasing use of electronic records in commerce, government, health and other fields has led to public fears about the potential mis-uses of private data. Once personal information has been submitted to an organisation, the subject of that information no longer controls what becomes of it, and organisations or rogue parties within organisations have the potential to mis-use the information through negligence or dishonesty.

While some organisations publish privacy protection policies, there is no technological guarantee that the policy espoused by the organisation will actually be followed by the people who have access to personal information. Furthermore, the privacy policies offered by organisations may not always meet the requirements or desires of the individuals who are the subjects of personal information held by those organisations.

Digital rights management ("DRM") provides protection for information by making access to information depend on satisfying the conditions imposed by a *licence* 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 [14] by developing licences that represent individuals' preferences for use of their personal information. The digital rights management approach to privacy is detailed in Section 2.

The MPEG-21 Multimedia Framework [9] is a framework for creating, distributing, navigating, using and controlling multimedia content, currently under development by the Motion Picture Experts Group ("MPEG"). Of particular interest to this paper, MPEG-21 proposes to incorporate an *Intellectual Property Management and Protection* framework within which content providers can control the use and distribution of multimedia content. In this paper, we consider that "multimedia content" might include personal information such as contact details and financial records. We will give an outline of the relevent components of the MPEG-21 Framework in Section 3.

The MPEG Rights Expression Language supplies a vocabulary of elements useful in copyright protection applications, but lacks elements that are useful in privacy protection applications. In Section 4, we outline how we developed a "privacy extension schema" (in the sense of XML Schema) for MPEG REL, based on a study of vocabularies developed for the Platform for Privacy Prferences [22] and Enterprise Privacy Authorization Language [19]. Our extensions allow individuals to express how they allow their data to be used in terms of actions and conditions that can be interpreted by an automated computer terminal.

In Section 5, we describe the extension of an existing MPEG-21-based digital rights management system to a privacy protection scenario. Our implementation allows a service provider to collect individuals' data in the form of XML documents, while the use and distribution of these documents is restricted according to conditions supplied by the data's owner.

Our system demonstrates the fundamentals of the DRM approach to privacy, but leaves substantial opportunity for further work in a number of areas including the composition of licences, management of protected information and provision for exceptional circumstances. We will conclude the paper with a discussion of outstanding issues in Section 6.

2 Digital Rights Management and Privacy Protection

Zittrain [24] observed a duality between protection of private data, and protection of copyrighted material: in both cases, we have a provider who wishes to make some information available to a third party in return for some financial reward or service, but does not wish to make the information publicly available. Technical approaches to protecting copyright, therefore, might be expected to yield insights into technical approaches to protecting privacy.

Kenny and Korba [14] later examined applying digital rights management technology in the context of the European Union's Data Protection Directive. Unlike models of privacy protection in which the privacy policy is developed by the database operator, the digital rights management model permits the data subject to choose the policy to be applied to his or her data.

Figure 1 shows the architecture of a typical digital rights management system. Data is created by a *provider*, and transmitted in a protected (for example, encrypted) form to a *user* via some distribution channel. In order to access the protected data, the user must obtain a *licence* from the licence issuer. A licence is

a document containing the terms of use of the data and the information required to access the protected content.
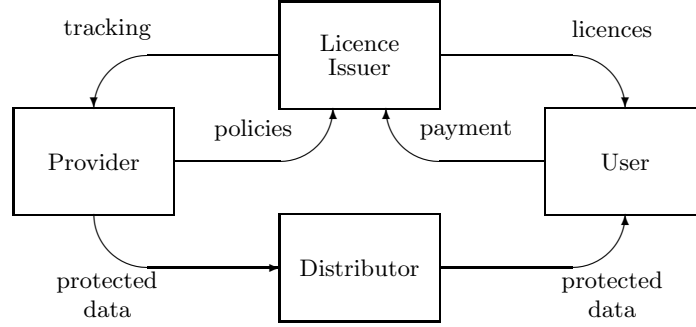

Fig. 1. A typical digital rights management system

Protected data may only be accessed using special terminals that are certified to behave in accordance with the terms specified in the licence, and not to reveal the unprotected content or decryption keys to the human user of the terminal. By controlling the licences that are made available to users (via the licence issuer), the provider can control whether or not content is copied between users, how many times content is used, and so on.

In a privacy protection context, the provider is a *data subject* whose privacy is at stake should an item of data be mis-used in some way. A *data user* may require access to the data for some purpose, such as completing a transaction requested by the data subject. In order to gain access to the data, the data user must obtain a licence from the licence issuer. Licences issued by the licence issuer are controlled in some way by the data subject, either directly or by having the issuer act according a policy supplied by the data subject. The data user can then access the data according to the terms of the licence.

Several DRM-like approaches to privacy protection have been reported in the literature, often for specific applications such as location privacy. We are not aware of any attempt, however, to develop a privacy protection system in the digital rights management model as complete as those currently available for copyright protection.

Cha and Joung's *On-Line Personal Data Licensing* ("OPDL") system [3] allows data subjects to issue licences using a *personal data licensor*. The personal data licensor is much like the licence issuer in a digital rights management system. OPDL licences are based on the policy language defined by the Platform for Privacy Preferences ("P3P") [22]. P3P, however, was not designed for this purpose and does not provide for automated enforcement of the policies expressed in its policy language. In OPDL, licences are simply stored and made available to any audit of the privacy practices of the data collector.

Hong and Landay's *Confab* architecture [8] for ubiquitous computing allows items of data to be associated with a "privacy tag" (licence). The privacy tag specifies the conditions under which the data may be retained, and provides an e-mail address to which notifications of disclosure can be sent. The tag does not specify how the data may be used or shared, however.

The most similar system to the one described in this paper is the *Personal DRM* ("PDRM") system of Gunter, et al. [7]. PDRM is a location privacy system in which individuals may make their current location available in order to receive some service, such as alerting them to the proximity of their friends. Individuals' privacy is protected by associating their location data with a licence written in the Extensible Rights Markup Language ("XrML") [4], which is the predecessor of the language used by MPEG-21 and in this paper.

PDRM's XrML, however, makes extensive use of P3P policy files to describe users' privacy preferences and Gunter, et al. do not describe any method by which these preferences can be enforced. For the system described in the present paper, we developed a rights expression language within the model used by both XrML and MPEG-21 that can be enforced using the standard algorithm for interpreting these languages.

3 MPEG-21

Unlike the well-known MPEG-1, -2 and -4 standards, MPEG-21 does not define the way in which individual multimedia presentations are encoded, but defines ways in which atomic multimedia objects can be used, combined, navigated and referenced. It consists of numerous parts, some of which have been ratified by the International Standards Organisation as the ISO/IEC 21000 series of standards, while others remain under development. In this section, we will give an overview of the components of MPEG-21 required to understand this paper.

3.1 Digital Items

The core notion in MPEG-21 is the notion of a *digital item* [10], which represents a collection of multimedia objects related in some way. Digital items are described using the XML-based *digital item declaration language* ("DIDL"), which organises content and meta-data into a hierarchical structure. For the purposes of this paper, the most important elements are:

Resources. Atomic multimedia objects such as images, sounds and videos.
Components. Resources together with their descriptors.
Descriptors. Meta-data, such as identifiers, MPEG-7 descriptors, etc.

Figure 2 shows a simple digital item declaration, similar to the digital items used in our system. It consists of a single item containing a single component. The resource is an XML document contained by the *MyXML* tags (the body of the document has been omitted for brevity), and is identified by the URN `urn:smartinternet:doc1`.

```
<didl:DIDL>
  <didl:Item>
    <didl:Component>
      <didl:Descriptor>
        <didl:Statement>
          <dii:Identifier>urn:smartinternet:doc:1</dii:Identifier>
        </didl:Statement>
      </didl:Descriptor>
      <didl:Resource>
          <myxml:MyXML>...</myxml:MyXML>
      </didl:Resource>
    </didl:Component>
  </didl:Item>
</didl:DIDL>
```

Fig. 2. A simple digital item declaration.

3.2   Intellectual Property Management and Protection

Intellectual property management and protection ("IPMP") is MPEG's term for digital rights management [11]. MPEG-21 does not define a digital rights management system in the same way that it defines digital item declarations, digital item identifiers and so forth, 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.

We say a resource is *governed* if it is protected by one or more IPMP tools. Each governed resource is associated with a plaintext identifier and an *IPMP information descriptor* that associates the resource with a licence and describes the IPMP tools required to access the resource. If the conditions of the licence are satisfied, the terminal must obtain and instantiate the IPMP tools in order to access the resource.

A large part of the work done on our original digital rights management system involved the design and implementation of IPMP tools. The security architecture used by our tools is described in Appendix B, but the technical detail of their implementation is beyond the scope of the present paper.

3.3   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" [12]. MPEG REL is closely based on the Extensible Rights Markup Language ("XrML") [4].

An MPEG REL licence is structured as a collection of *grants* issued by some licence issuer. Each grant awards some *right* over some specified *resource* to a 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 licence 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 schemata, 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`). These schemata define the fundamental elements of the language, some widely-useful conditions, and elements useful in copyright protection applications, respectively. We will later discuss the development of a *privacy extension schema* for use in privacy protection applications. We will denote elements of the privacy extension schema by the namespace prefix `px`.

Figure 3 shows an example of an MPEG REL grant allowing a principal (`r:keyHolder`) identified by his or her public key to print a resource (`mx:diReference`) identified by a digital item identifier `urn:smartinternet:doc1`. The principal is only permitted to print the resource once (`sx:ExerciseLimit`).

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

Fig. 3. An MPEG REL grant.

XrML and MPEG REL are provided with a vocabulary useful in copyright protection applications. In the following section, we will discuss extending MPEG REL with a vocabulary suitable for privacy protection applications.

4 Expressing Privacy Preferences in MPEG REL

As described in Section 2, previous authors have attempted to enlist the P3P policy language for expressing the privacy preferences of data subjects. The in-

tention of P3P, however, is to inform data subjects of the global privacy practices of Internet service providers. Here, we require data subjects to specify their preferences regarding the handling of a particular item of data. P3P seems poorly suited to the latter task since it provides no way of identifying a specific item of data or a specific data user. Furthermore, P3P does not provide for automated enforcement of privacy policies and we are not aware of any algorithms for determining whether or not a given action is permissible, given a P3P policy.

Recognising the shortcomings of P3P as an enforcement tool, researchers at IBM proposed the Enterprise Privacy Authorization Language ("EPAL") [19]. EPAL is intended to express an organisation's privacy policy in such a way as to make it enforceable by an access control system. EPAL's structure is very similar to that of MPEG REL and other access control languages such as the Extensible Access Control Markup Language ("XACML") [17]: policies in all of these languages consist of a series of rules expressing the right of some actor to perform some action on some object, subject to certain conditions and obligations. EPAL has an additional element called *purpose* that makes permission conditional on the action being performed for some particular purpose.

EPAL and XACML, however, require each organisation to define its own vocabulary of actors, actions, etc. for use in their access control policies. In our application, it seems highly impractical to require data subjects to use a different vocabulary for every service provider with which he or she interacts.

MPEG REL is specifically designed for the digital rights management model, provides a vocabulary that is constant across all service providers, and specifies an algorithm for determining whether or not a given action is permissible. However, the existing MPEG REL vocabulary was designed with only copyright protection applications in mind and it lacks elements to describe principals, rights and conditions that may be useful in privacy protection applications. For example, privacy protection systems often restrict the use of data to a particular transaction, but MPEG REL does not define any conditions that support this.

For the purposes of the prototype system described in Section 5, we designed a preliminary privacy extension schema by examining existing vocabularies for P3P (including drafts of P3P Version 1.1) and EPAL [18]. The detailed syntax of the extension was worked out by attempting to write licences for a variety of simple scenarios, and making corrections as necessary until the licences we wanted could be written reasonably conveniently. The resulting schema was applied to the customer service application considered in this paper.

The detailed development of a comprehensive privacy extension schema is left as the topic for another paper. In this section, we will simply summarise the major observations we made while developing our schema. A summary of the schema we derived is given in Appendix A.

Purposes. Perhaps the most conspicuous difference between MPEG REL and languages developed in privacy protection is the latter's use of "purposes". Different languages make somewhat different uses of the term – P3P Version 1.1 even goes so far as to use the term twice (once as "purpose" then again as "primary purpose") with different meanings. In the MPEG REL model, a "purpose"

must be interpreted as some combination of a particular principal exercising a particular right under certain conditions.

P3P's notion of a "purpose" generally corresponds to a combination of a right and one or more conditions in MPEG REL. For example, P3P's `contact` and `telemarketing` purposes can be interpreted as the right to contact someone, under the condition that it not be by telephone or be by telephone, respectively.

The use of "purpose" in EPAL at first recalls a condition in MPEG REL, and of course it would be possible to simply create an MPEG REL condition called `Purpose` that made a grant available only if the right was exercised for some specified purpose. This is, in fact, how purposes are treated in the Privacy Policy Profile of XACML Version 2.0. In our schema, we chose to create a different condition for every purpose that we interpreted in this way – this makes the vocabulary of purposes shared by all uses of the language.

However, a number of the purposes identified in [18] may be better implemented as roles in the sense of a role-based access control system. For example, it is much more straightforward to check that a principal is acting in the role of a police officer than it is to check directly that he or she is carrying out law enforcement. MPEG REL supports role-based principals using the `PropertyPossessor` principal (e.g. a principal who possesses the property of being a police officer). We will give more detail about how these elements are used in Section 5.

Obligations. EPAL and XACML distinguish "conditions" and "obligations" that represent conditions that must be true before access is permitted, and actions that must be carried out after access is permitted. MPEG REL conflates obligations with conditions – we can think of obligations as being post-conditions and EPAL/XACML-style conditions as being pre-conditions. It is straightforward to express widely-used obligations involving notification and data retention, for example, using `TrackReport` and `ValidityInterval` conditions in MPEG REL.

Recipients. P3P and [18] consider "recipients" who have data disclosed to them by someone with direct access to the database, but who do not have direct access to the database themselves. In the model used by P3P and EPAL, it makes sense to make the discloser to be the principal of an access control rule and make the identity of the recipient a condition. In the digital rights management model, however, it makes more sense to identify the recipient as the principal of a grant that is given directly to that recipient. The "discloser" can give the data to the recipient in its protected form without needing to access the data him- or herself.

5   Enforcing Privacy Preferences with SITDRM

In order to explore the digital rights management approach to privacy protection, we applied our existing implementation of MPEG-21's IPMP Components – known as "SITDRM" – to a privacy protection scenario.

SITDRM was designed to allow businesses to license multimedia works from their web site, using the MPEG-21 IPMP framework to ensure that buyers complied with the terms of the licence they had purchased. In the project described

by this paper, we took the IPMP technology that underpins SITDRM and applied it to the protection of customer records submitted via a company's web site. We call the new system "SITDRM Enterprise".

Figure 4 shows an overview of the SITDRM Enterprise system. We assume that some service provider (the data controller) requires information to be collected from its customers (data subjects), and that all of this information is stored in some central database. The service provider's employees (data users) then require access to the information in order to carry out their jobs and provide service to the customers.
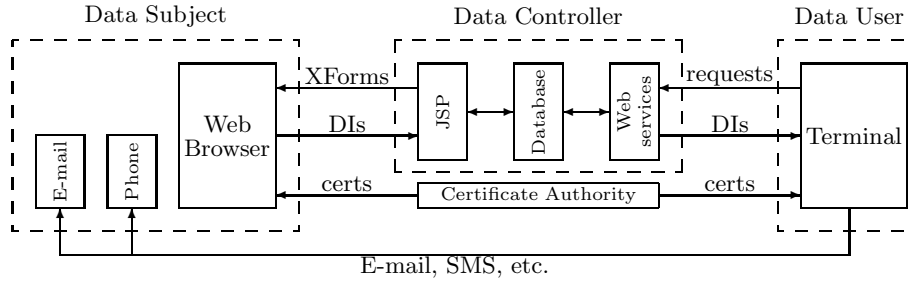

Fig. 4. SITDRM Enterprise Architecture

Customers submit their information via a form on the service provider's web site. In our example, the document contained the customer's postal address, e-mail address and telephone number formatted as an XML document using the Extensible Customer Information Language ("xCIL") [16]. In principle, the service provider can set up the web site to collect any information formatted as an XML document. At the same time, customers may design an MPEG REL licence 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 the licence designed by the customer. The governed item and issued licence 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 licence that authorises this action. If an appropriate licence is found, the action is permitted to continue. Otherwise, the action is rejected.

In general, governed items and licences can also be passed on to third parties (such as related companies) via e-mail or the like. If the customer has granted a licence that permits the third party to access his or her data, the third party can access this data as for employees of the original service provider. Our initial scenario considers data distributed within one company only, however.

5.1 Security Architecture

SITDRM Enterprise uses the same techniques used to preserve the integrity of the digital rights management as were used in the original copyright protection application. Our fundamental requirement is that every terminal be tamper-resistant and be supplied with a public/private key pair of which the private key is known only to the terminal – in particular, it is not known to the human user of the terminal. We further assume that a public key infrastructure exists that allows the public keys of terminals and licence issuers to be verified.

Every governed resource is encrypted using a unique *resource key*. Any licence that grants permission to use this resource must contain the resource key encrypted either by the public key of the terminal on which the resource is to be used, or by a key that can be obtained from a second licence without which the first licence would be invalid. In this way, a resource can only be decrypted by a tamper-resistant terminal in possession of a valid set of licences. The integrity of licences is ensured by having them signed by their issuer.

For clarity of the main body of this paper, we have omitted the details of cryptographic operations in the remainder of this section. A complete description of SITDRM's security architecture is given in Appendix B.

5.2 Licences

Two kinds of licences are used in SITDRM Enterprise: *Membership certificates* permit individual data users to act as members of roles using the `PossessProperty` right, while *resource licences* permit members of roles to perform actions using the `PropertyPossessor` principal.

In order for a particular data user to carry out an action on a document, he or she must obtain both a resource licence that permits some role to carry out that action, and a membership certificate that makes him or her a member of that role. Examples of a membership certificate and a resource licence are given in Figures 5 and 6, respectively.

```
<r:grant>
  <r:keyHolder>
    <r:info>
      <dsig:KeyValue>
        <dsig:RSAKeyValue>...</dsig:RSAKeyValue>
      </dsig:KeyValue>
  </r:keyHolder>
  <sx:possessProperty/>
  <sx:propertyUri definition="urn:smartinternet:customer-service"/>
</r:grant>
```

Fig. 5. A membership certificate for the `urn:smartinternet:customer-service` role.

```
<r:grant>
  <r:propertyPossessor>
    <sx:propertyUri definition="urn:smartinternet:customer-service"/>
  </r:propertyPossessor>
  <mx:play/>
  <mx:diReference>
    <mx:identifier>urn:smartinternet:customers:123</mx:identifier>
  </mx:diReference>
</r:grant>
```

Fig. 6. A licence that allows members of the `urn:smartinternet:customer-service` role to view the document `urn:smartinternet:customers:123`.

Membership certificates can be obtained from a *role issuer* operated by the service provider. The role issuer is simply a licence issuer in the sense described in Section 2. For our purposes, we assume that the role issuer is operated by some reputable administrator who is trusted to issue membership certificates only to individuals who have reason to act in those roles. In a real company, we might expect the role issuer to be under the control of human resources staff who assign roles to employees according to the terms of their employment.

Resource licences are issued by data subjects. The details of generating and issuing appropriate licences will be discussed in Section 5.3.

5.3 Submitting Governed Documents

A service provider who wishes to collect information from his or her customers may design a form for doing so using XForms [23]. XForms' ability to manipulate arbitrary XML documents without programming makes it very appealing to web designers who need to present documents written in machine-oriented languages such as xCIL and MPEG REL in way that is accessible to human users.

Every XForms form is associated with an XML document called the *instance document*. Every control on the form is identified with a node of the instance document using an XPath expression [21], and the user's input to a control determines the content of the associated XML node. Initial values for controls can be supplied by the data controller by supplying an initial instance document containing those values. When the user chooses to submit the form, the instance document is uploaded to the server.

The present application uses two kinds of instance documents: data and licences. We require that the former kind be encrypted before it is uploaded to the server, and that the latter kind be signed before it is uploaded. For this purpose, we added a new attribute to the `submission` element of XForms – called *transform* – that indicates what kind of post-processing should be applied to the instance document prior to uploading it. We use transformations called *govern* and *issue* that cause the instance document to be converted into an MPEG-21 governed digital item and issued as licence, respectively.

A simple form for editing a document and a licence is shown in Figure 7. Each `model` element in the head of the HTML page describes one instance document, and every control on the form is associated with a model using the `model` attribute. In the example, model `d` represents the document and model `l` represents the licence. The submit button and other details have been omitted or abbreviated for brevity.

```
<h:head>
  <f:model id="d">
    <f:instance src="/templates/document.xml"/>
    <f:submission action="/submission/document" transform="govern"/>
  </f:model>
  <f:model id="l">
    <f:instance src="/sitput/templates/licence.xml"/>
    <f:submission action="/submission/licence" transform="issue"/>
  </f:model>
</h:head>

<h:body>
    <f:input model="d" ref="/ci:xCIL/.../ci:ContactNumber">
      <f:label>Phone Number</f:label>
    </f:input>
    <f:input model="l" ref="/r:license/.../px:ContactMethodUri/@definition">
      <f:label>Voice or SMS</f:label>
    </f:input>
</h:body>
```

Fig. 7. A form for editing a document and a licence.

The form in Figure 7 initialises the instance documents from templates on the server called `document.xml` and `licence.xml`. In our example application, the document template is a skeleton xCIL document whose fields will be filled in by the form controls.

The licence template, however, is a near-complete licence similar to the one shown in Figure 8. This template supplies technical information such as the identifier for the role that will be using the information, while allowing the data subject to change the permissible contact method using the form. Data subjects may view the complete technical details of the licence using a toolbar option.

In principle it is possible to design a form that allowed the data subject to make any change to the licence he or she wishes. Such a form, however, would likely be very intimidating to users and we expect that most users would only be interested in modifying a few simple conditions like the one shown in Figure 8. We will discuss this issue further in Section 6.

```
<r:grant>
  <r:propertyPossessor>
    <sx:propertyUri definition="urn:smartinternet:customer-service"/>
  </r:propertyPossessor>
  <px:contact/>
  <px:contactMethods>
    <px:contactMethodUri definition="changeme"/>
  </px:contactMethods>
</r:grant>
```

Fig. 8. A licence template for Figure 7.

5.4 Accessing Governed Documents

Anyone with access to the data controller is permitted to download any of the
documents and licences stored there. Documents and licences so obtained may be
further distributed using other channels, for example, by e-mailing them to other
companies or saving them to physical media. However, a governed document
can only be accessed on a DRM-compliant terminal and only if that terminal is
provided with licences that permit access to that document.

In our implementation, a DRM-compliant terminal is represented by an ap-
plication called "IPDoc" that allows users to download governed documents from
the server and perform actions on them if there are licences permitting them to
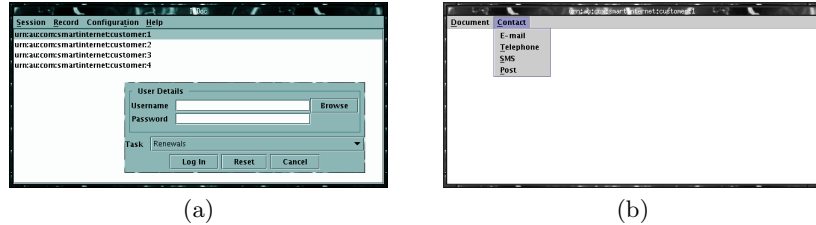do so. Some screenshots from IPDoc are shown in Figure 9.

(a)   (b)

Fig. 9. IPDoc: (a) the main window and log-in dialogue and (b) a document window.

IPDoc's main window lists the identifiers of all of the documents in the
database. Before any documents can be used, the user must log-in with a name
and password, and specify the task that he or she intends to perform – in this
case, either "renewals" or "marketing". Of course the latter selection is open to
abuse since the computer cannot check what the user actually intends, but it
serves to at least prevent honest users from using or disclosing data by mistake.

When the user selects a document from the main window, IPDoc downloads
the document from the database and opens a new window with menu options
for performing various actions on the document. If the user chooses to perform

an action on a document, IPDoc first searches for any licence that permits that action. If it finds one, and that licence requires the user to be a member of a particular role in order to be used, it then searches for a membership certificate that permits the current user to act in that role. If it finds one, the action is permitted. Otherwise the action is rejected and an error message is displayed.

6 Lessons Learnt and Future Work

Composing Licences. To be enforceable, licences must be expressed in terms of the internal structure and procedures of the service provider. Data subjects, however, are unlikely to find this representation very convenient or meaningful when attempting to express their privacy preferences.

Our XForms-based approach allows licences to be represented in a more convenient way by using careful web design, but is limited to making direct associations between form controls and MPEG REL licence elements. Nor does it assure a data subject that the licence being produced accurately represents their privacy preferences unless they have a detailed understanding of MPEG REL and the time to examine the licence.

Improving the way that licences are presented to users and giving data subjects greater assurance that licences match their preferences is the subject of further research. Possible approaches include auditing of web sites by consumer agencies, protocols for negotiating privacy policies [5, 13, 20] and the introduction of a formal human-readable representation of privacy preferences that can be mapped to computer-readable licences by machine.

Selecting Documents. Our implementation allows data users to select documents based on the identifier associated with the document. This may be acceptable if the identifiers used are meaningful, or if documents can be chosen automatically by a computer system that knows which document ought to be processed next (for example, by maintaining a queue of jobs to be done). However, we can imagine situations in which more useful information would be required in order for a user to decide which document is the one that he or she is looking for – for example, if a user were looking for documents concerning a particular topic.

DIDL allows meta-data to be associated with a resource by placing it in a `Descriptor` element contained within the component that contains the resource. This descriptor need not be encrypted even if the resource is governed, and can be used by a data user to identify resources that he or she might be interested in. Obviously, however, meta-data may itself constitute private information.

Possible solutions include the use of a trusted search engine [15] and encrypted keyword search [1]. These topics are beyond the scope of this paper.

Exceptions. Our system allows data to be used in any situation that can be foreseen by the data subject at the time the data is created. However, it is easy to imagine unforeseen exceptional circumstances – such as a medical emergency – in which it may be desirable to over-ride the restrictions imposed by a licence.

Even if a data subject could foresee all of the exceptional circumstances in which data might need to be accessible, it seems likely that encoding all of them into a licence would be cumbersome and inefficient. Furthermore, there may be cases (notably in law enforcement) where the data subject may not have any incentive to encode exceptions.

These exceptions can be considered loosely analogous to the fair dealing or fair use exceptions of copyright law, which allow content users to make some copies of copyrighted content without the explicit permission of the copyright owner. Dealing with these exceptions is very difficult [6], though some authors have proposed methods using a trusted escrow agent who is able to over-ride a DRM system if a case for an exception can be made [2]. The development of analogous systems for privacy is left as future work.

7 Conclusion

SITDRM Enterprise shows how a DRM framework originally developed for copyright protection can be applied to privacy protection. It shows how data subjects' preferences for the use of their data can be encoded in such a way as to enable a computer system to – so far as is possible using current technology – ensure that those preferences are adhered to by data users.

Compared to models in which private data is governed by a central policy set by the organisation's privacy officer, the digital rights management model permits data subjects to control the policy to which their data is subject and ensures that this policy is applied in any organisation to which the data might travel. The need to compose, manage and interpret large numbers of licences, however, makes the system somewhat more complex than one in which all data is subject to a central policy. In particular, the average user may require technological assistance to be able to produce useful and accurate licences conveniently.

In designing SITDRM Enterprise, it quickly became apparent that the architecture we had designed might work just as well for protecting internal documents generated by company employees as it does for protecting external documents submitted by data subjects. One might wonder if it is possible to develop a "grand unified rights management system" that could be deployed in any application where there are rights to be protected. Our work with SITDRM may suggest that this is possible, but it remains to be be seen whether or not a unified rights management system could be as practical and effective as one designed for a specific purpose.

8 Acknowledgements

References

1. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.
2. D. L. Burk and J. E. Cohen. Fair use infrastructure for copyright management systems. *Harvard Journal of Law and Technology*, 15:41–83, 2001.
3. S.-C. Cha and Y.-J. Joung. From P3P to data licenses. In *Workshop on Privacy Enhancing Technologies*, pages 205–221, 2003.
4. ContentGuard. Extensible Rights Markup Language. `http://www.xrml.org`, 2004.
5. K. El-Khatib. A privacy negotiation protocol for web services. In *Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments*, 2003.
6. J. S. Erickson and D. K. Mulligan. The technical and legal dangers of code-based fair use enforcement. *Proceedings of the IEEE*, 92:985–996, 2004.
7. C. A. Gunter, M. J. May, and S. G. Stubblebine. A formal privacy system and its application to location based services. In *Workshop on Privacy Enhancing Technologies*, pages 256–282, 2004.
8. J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *International Conference On Mobile Systems, Applications And Services*, pages 177–189, 2004.
9. International Standards Organisation. Information technology – multimedia framework (MPEG-21) – part 1: Vision, technologies and strategy. ISO/IEC 21000-1:2001.
10. International Standards Organisation. Information technology – multimedia framework (MPEG-21) – part 2: Digital item declaration. ISO/IEC 21000-2:2003.
11. International Standards Organisation. Information technology – multimedia framework (MPEG-21) – part 4: Intellectual property management and protection components. ISO/IEC 21000-4:2006.
12. International Standards Organisation. Information technology – multimedia framework (MPEG-21) – part 5: Rights expression language. ISO/IEC 21000-5:2004.
13. K. Irwin and T. Yu. Determining user privacy preferences by asking the right questions: An automated approach. In *ACM Workshop on Privacy in the Electronic Society*, pages 47–50, 2005.
14. S. Kenny and L. Korba. Applying digital rights management systems to privacy rights. *Computers & Security*, 21:648–664, 2002.
15. E. Mykletun and G. Tsudik. Incorporating a secure coprocessor in the database-as-a-service model. In *International Workshop on Innovative Architecture for Future Generation High Performance Processors and Systems*, pages 38–44, 2005.
16. Organization for the Advancement of Structured Information Standards. OASIS Customer Information Quality TC. `http://www.oasis-open.org/committees/ciq/`, 2004.
17. Organization for the Advancement of Structured Information Standards. OASIS eXtensible Access Control Markup Language TC. `http://www.oasis-open.org/committees/xacml/`, 2004.
18. C. Powers, S. Adler, and B. Wishart. EPAL translation of the Freedom of Information and Protection of Privacy Act. White paper, Ontario Information and Privacy Commissioner, 11 March 2004. `http://www.ipc.on.ca/docs/EPAL%20FI1.pdf`.
19. M. Schunter and C. Powers. The Enterprise Privacy Authorization Language (EPAL 1.1). `http://www.zurich.ibm.com/security/enterprise-privacy/epal`, 2003.

20. A. Tumer, A. Dogac, and I. H. Toroslu. A semantic based privacy framework for web services. In *Proceedings of WWW '03 Workshop on E-Services and the Semantic Web*, 2003.
21. W3 Consortium. XML Path Language (XPath). `http://www.w3.org/TR/xpath`, 1999.
22. W3 Consortium. Platform for Privacy Preferences (P3P) project. `http://www.w3.org/P3P`, 2004.
23. W3 Consortium. XForms. `http://www.w3.org/MarkUp/Forms`, 2005.
24. J. Zittrain. What the publisher can teach the patient: Property and privacy in an era of trusted privication. *Stanford Law Review*, 52, 2000.

A A Privacy Extension Schema for MPEG REL

The multimedia extension of MPEG REL provides methods of identifying an item of content that seem sufficient for privacy protection applications, and so there does not appear to be any need to introduce new kinds of resource in our privacy extension.

The MPEG REL core schema provides methods of identifying roles and individuals that seem sufficient for privacy protection applications. However, it seems useful to allow the destination of a transfer right (such as `Embed`) to be a database or other object that is an MPEG REL resource in its own right. Since the syntax of the `Destination` condition requires the destination to specified as a principal, we must introduce a new principal – which we call `ResourcePrincipal` – that makes a resource into a principal. (Of course, we could also modify the syntax of the `Destination` condition in the multimedia extension schema.)

Table 1 lists the new rights that we identified for inclusion in our privacy extension schema, and Table 2 lists the conditions. For the most part, these are derived by decomposing the "purposes" and "primary purposes" of P3P into a combination of an action and the conditions under which that action may take place. Of course, a number of the actions and conditions so derived are already present in the standard extension and multimedia extension schemata, and we have not duplicated such elements in our privacy extension schema.

Right	Description
`Contact`	Use the resource to establish a communications channel
`Export`	Export the resource to an ungoverned application or database
`Query`	Submit the resource as a query to a service
`Tailor`	Use the resource for a transient adaptation of a second resource

Table 1. Rights in our privacy extension schema.

Note that the `Export` right is present in XrML, but not in MPEG REL. This right seemed to us to be necessary for allowing data to be exported to a specific application or database that performed some function that lay outside

the domain of a terminal of the kind postulated by MPEG-21. The `historical` purpose of P3P, for example, contemplates data being exported to some historical archive. It is unlikely, however, that such an archive would be maintained by a terminal like IPDoc.

Condition	Description
`ContactMethod`	Only if the specified means of contact is used
`Dealing`	Only in the context of a particular session or goal
`Pseudonym`	Only if the data is anonymised or pseudonymised

Table 2. Conditions in our our privacy extension schema.

A number of "primary purposes" used in P3P Version 1.1 suggest the use of a `Content` condition that restricts the kind of material present on a communications channel to news, entertainment, marketing, etc. We are not aware of any computer system that can vet the contents of a channel in this way and so have chosen not to include such a condition in our privacy extension schema. Restrictions of this sort can be achieved to some degree using the `Dealing` condition, however, as demonstrated in our example scenario described in Section 5.

B   Security Architecture

In order to preserve the integrity of the digital rights management system, governed content must only be usable under the terms imposed by a licence supplied by the licence issuer. To this end, we require that

– content may only be accessed by use of a secure terminal trusted to comply with the terms of any licence associated with the content; and
– terminals must be able to verify the authenticity and integrity of any licences purporting to grant privileges over content.

B.1   Key Infrastructure

We assume that every trusted terminal T has a private key $\bar{K}_T$ and corresponding public key K_T, and that the authenticity of the public key K_T can be verified by the licence issuer using some public key infrastructure. 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. In our implementation, we use the well-known RSA algorithm for all public key cryptographic operations.

We similarly assume that every human user u (both data subjects and data users) of the system has a private key $\bar{K}_u$ and public key K_u. This key pair is used both for identifying the beneficiary of a licence using the MPEG REL `KeyHolder` principal, and for signing licences issued by data subjects. We also

assume that every human user has a secret symmetric *master key k_u* that will be used for encrypting his or her data according to an algorithm described below.

Finally, each role R is associated with a key pair $\bar{K}_R$ and K_R that is used for encrypting keys to be delivered to that role. We assume that the public key for all of the roles in the system can be obtained from the certificate authority.

B.2 Resource Encryption

Every document x to be submitted to the data controller must be encrypted with a unique resource key k_x. In order to generate a unique resource key, we require every document x to be associated with a unique digital item identifier i_x. A unique resource key is then generated according to the formula

$$k_x = \text{HMAC-SHA1}(k_u, i_r)$$

where k_u is the master key of the user who created the document. We use the AES algorithm for all symmetric encryption.

In SITDRM Enterprise, uniqueness of resource identifiers is ensured by assuming that every data controller is associated with a unique URI stem. Every time the data submission form is downloaded from the web server, the data controller uses a counter to generate a new suffix to its URI stem. In our example, the data controller was assigned the stem `urn:au:com:smartinternet` and documents are numbered `urn:au:com:smartinternet:customer:1`, `urn:au:com:smartinternet:customer:2`, etc. in the order in which they are submitted.

B.3 Licences

In SITDRM, every grant of a licence that permits some action to be performed must contain the key required to perform that action. For security, the key must be encrypted in such a way as to render it inaccessible to any party except one that is entitled to perform the action.

Every resource licence is required to contain the resource key for the resource to which it refers, encrypted by the public key of the role to which that licence is awarded. In order to access the resource key, the private key of the role must be obtained from a membership certificate for that role.

Since data users are not assumed to be trusted, it is not sufficient to encrypt the private key of a role using the public key of the data user for whom a membership certificate is intended – this would allow a dishonest data user to obtain the resource key for a resource. Instead, we require that membership certificates only be usable on a particular terminal, that is, that a data user may only act as a member of the role when he or she is using a particular terminal (presumably one that is owned and operated by the data user's employer).

The private key of a role is encrypted using the public key of the terminal on which the membership certificate is to be used, and inserted into the membership certificate. In this way, the terminal can decrypt the role's private key from the membership certificate and use this in turn to decrypt the resource key in a

resource licence. The terminal is trusted not to reveal the role's private key, the resource key or the decrypted resource to its human user.

Membership certificates are signed by the role issuer. We assume that a trusted version of the role issuer's public key can be obtained from the certificate authority. Any terminal can then verify the integrity of a membership certificate by verifying the signature of the role issuer on that certificate.

Unfortunately, the same approach does not suffice for resource licences. Since all of the humans who use the system have the ability to issue resource licences, it is possible for a dishonest user to issue a licence for a document created by any data subject. This can be done by copying the encrypted resource key and encrypted resource into an arbitrary licence, and signing this licence using the dishonest user's private key. The forged licence will be accepted as valid by the terminal for which the original licence was intended.

There is a fairly simple fix for this problem, though we have not yet implemented it in SITDRM Enterprise. The strategy is to insert a secret into both the encrypted resource and the signed licence, such that the terminal is able to recover the secret from both (using its private key) and check that they match. An attacker is then unable to generate a valid signature for a licence on this resource since he or she is unable to insert the secret.

Let n_x be a random nonce chosen by the data subject every time he or she encrypts a document x. The nonce is appended to the document prior to encryption. That is, the encrypted document is $\hat{x} = e(k_x, x \parallel n_x)$ where $e(k, m)$ denotes symmetric encryption of message m with key k and $\parallel$ denotes concatenation.

Let E be a public key encryption algorithm and S be a signature algorithm, using parameters analogous to e above. The data subject u can compute a signed licence $\hat{L}$ as follows:

1. Compute $\hat{k}_x^* = E(\bar{K}_u, n_x \parallel k_x)$, that is, the nonce and content key encrypted using the private key of the data subject.
2. Compute $\hat{k}_x = E(K_T, \hat{k}_x^*)$, that is, the nonce and content key further encrypted using the public key of the terminal.
3. Compute $\sigma = S(\bar{K}_u, L \parallel \hat{k}_x \parallel K_u)$, that is, the data subject's signature on the original licence L and encrypted nonce and content key.
4. Compute the signed licence $\hat{L} = L \parallel \hat{k}_x \parallel K_u \parallel \sigma$.

A terminal can then verify the signature on such a licence as follows:

1. Check that σ is a valid signature for $L \parallel \hat{k}_x \parallel K_u$. If not, stop.
2. Decrypt $\hat{k}_x$ using $\bar{K}_T$ to obtain $\hat{k}_x^*$.
3. Decrypt $\hat{k}_x^*$ using K_u to obtain n_x and k_x.
4. Decrypt $\hat{x}$ to obtain n_x and x. If the n_x obtained from $\hat{x}$ is not the same as that obtained from $\hat{L}$, stop.

It is straightforward to check that the algorithm is both correct and secure, assuming that the encryption algorithm E and signature algorithm S are secure.