A Secure Electronic Healthcare Record Infrastructure in the Digital Rights Management Model

Sheppard, Nicholas; Safavi-Naini, Reihaneh; Jafari, Mohammad

A Secure Electronic Healthcare Record Infrastructure in the Digital Rights Management Model

Nicholas Paul Sheppard, Reihaneh Safavi-Naini and Mohammad Jafari
iCore Information Security Lab
Department of Computer Science
University of Calgary

Contents

Chapter 1

Introduction

Electronic healthcare record systems promise to increase the efficiency and effectiveness of healthcare systems by ensuring that healthcare workers can get timely access to the correct and complete information that they require in order to provide good health services to their patients. Electronic healthcare systems have been investigated in many countries, and numerous research journals and conferences are devoted to their design and evaluation.

Greater distribution of information through an electronic healthcare system brings with it a risk that patients' information will be misused, resulting in invasions of privacy and/or unfair discrimination on the basis of patients' medical histories. Security and privacy therefore forms an important part of any electronic healthcare system, and numerous designs for security and privacy in the healthcare space have been proposed over the years [4, 5, 10, 15, 18, 19, 20, 21, 23, 43, 45, 50].

Systems for controlling access to sensitive information, both in a healthcare context and others, are typically designed to enforce the *principle of least privileges*, that is, the principle that the human users of a system should have access to the minimum amount of information required to carry out their assigned job. This principle aims to minimise the potential for information to misused, without interfering with people's ability to do their jobs.

In a privacy context, the principle of *consent* is widely used in privacy law to restrict the disclosure of sensitive information according to the wishes of the subject of that information. *Electronic consent* (often shortened to "e-consent"), in particular, allows the subject of some electronic information to permit or deny the disclosure of that information to particular people in particular circumstances [12]. Electronic consent systems have been proposed as a method of controlling the disclosure of electronic healthcare records [3, 34, 35, 44, 49, 53], and (less frequently) for other kinds of personal information in electronic commerce contexts [6, 25, 28].

Electronic consent systems bear some resemblance to *digital rights management systems*. Digital rights management is best known for its use in the protection of intellectual property [31], but more recently has also been applied to the protection of personal information [26, 47]. Digital rights management technology allows information owners to control the distribution and use of their information by describing a

policy in a machine-readable *licence*. Information is distributed in a protected form such that it can only be accessed by special *DRM agents* that are trusted to comply with the terms specified licences.

Petković, et al. examine the potential for digital rights management technology in securing electronic healthcare records [40]. They argue that digital rights management technologies already provide many of the features desired in a secure electronic healthcare system, in that they can provide persistent and homogeneous protection of information even when it is disseminated throughout a distributed healthcare system.

However, they additionally identify a number of points on which existing digital rights management systems (specifically, those originally designed for managing the distribution of sensitive documents within corporate enterprises) do not meet these needs, including:

- the parties that access and manipulate documents may come from many different domains and it is difficult to predict in advance who these parties might be;
- the ownership of data is not clearly defined, as it is shared between healthcare workers and patients;
- access rights are highly context-dependent and are difficult to determine automatically (for example, is a request an emergency?);
- small fragments of records (and not just whole documents, as is usually the case in intellectual property protection) may be critical;
- the membership of roles can change very quickly;
- healthcare data may be used for research purposes in an anonymised form; and
- healthcare data is prone to numerous inference channels.

In the present document, we describe one possible implementation of a secure electronic healthcare infrastructure modelled on the digital rights management approach to privacy protection [26, 47] and workflow-based access control [2, 24, 45]. Our proposal attempts to address several of the points identified by Petković, et al., as well as other issues identified by our own research.

While many of the features of the proposed system could also be provided by an access control system and/or electronic consent system such as those proposed in earlier work, the proposed system additionally allows for

- persistent protection of information throughout the global electronic healthcare record infrastructure, local healthcare facilities and mobile healthcare workers;
- highly expressive consent directives that can be enforced in an automated fashion; and
- information flows that cross organisational boundaries.

Anonymisation and inference channels may additionally be addressed by other work in the iCore Information Security Lab.

In addition to our general application of digital rights management in a healthcare context, we introduce some new techniques with wider applications in digital rights management and access control, including

- the use of workflow information to provide fine control over the purposes for which rights-managed data is used; and
- the ability to transfer the execution of a task from one device to another (known as *session mobility* [46]) within the confines of a digital rights management system.

1.1 Document Outline

Chapter 2 gives a general overview of the proposed system suitable for readers with a modest technical background. Non-technical readers may also find Appendix A of interest. The remaining chapters present the detailed design of the system and are intended for a more technical audience with an interest in implementing, analysing and evaluating the proposed system.

Chapters 3 through 7 give a technical description of each of the components of the system in turn. Chapter 8 then outlines the technical procedures and protocols used for achieving particular tasks within the system. Chapter 9 describes the security properties of the system.

Finally, Appendix A describes an example application of the proposed system in a simple healthcare facility, and Appendix B gives the technical detail of some example authorisation policies that might be used in the system.

Chapter 2

Overview

The system proposed in the present document can be seen as a two-stage digital rights management system, composed of

- a global[1] rights management system that controls the distribution of healthcare information to healthcare facilities; and
- a facility-level rights management system that controls the distribution of information within facilities.

Patient consent and jurisdictional requirements are expressed in broad terms at the global level, and these are translated into specific terms in the context of a particular facility.

In this chapter, we will first give an introduction to digital rights management, then give an overview of each level of the proposed system in turn. We will give detailed descriptions of each component of the system in the remaining chapters of this document.

2.1 Digital Rights Management

Digital rights management has many similarities to traditional access control systems, but requires that information remain protected even when transported beyond the boundary of systems controlled by the information owner. Digital rights management can thus be defined as "persistent access control" [1], as distinguished from traditional access control systems that cannot (technologically) compel users to conform to any particular usage policy once they have been granted access to a piece of information.

Digital rights management allows protected information to be transmitted over an insecure channel and stored on an insecure storage device without compromising the integrity and confidentiality of the information. For example, information can be distributed via a direct network connection, a file-sharing network, or by copying it onto

[1] in practice, national or provincial

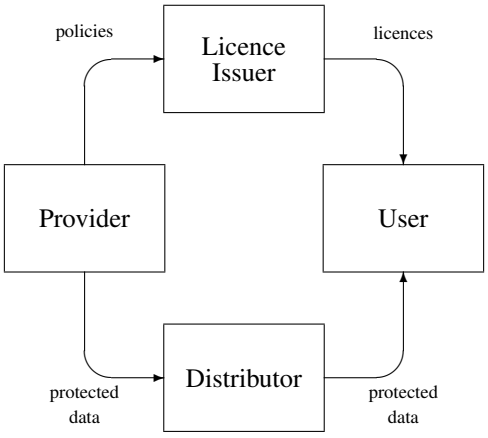

Figure 2.1: The components of a digital rights management system.

transportable media; and stored on a file server, an individual computer's hard drive, or removable media.

Figure 2.1 shows our reference model for a digital rights management system [31]. Information 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*.

Licences are written in a machine-readable *rights expression language* that sets out the terms of use of the data and the information required to access the protected content.

The fundamental security requirement for a DRM system is that the hardware and/or software used to access protected data be guaranteed by its manufacturer to behave in accordance with licences; it effectively performs the role of the "reference monitor" in traditional access control systems. For the purposes of this document, a *DRM agent* is an abstract single-user viewer, editor, or similar that may be implemented as a hardware device, a software application or combination of the two.

2.2 Global Secure Electronic Healthcare Infrastructure

Figure 2.2 shows a high-level overview of the whole secure electronic healthcare record system proposed in this document. It consists of

- an arbitrary number of *patients* (only one of whom is shown in the diagram);
- an arbitrary number of *healthcare facilities* (only three of which are shown) in which patients may seek treatment;
- a *consent directive management system* ("CDMS") that stores patients' *consent directives*, which record patients' consent (or not) to use information about them; and

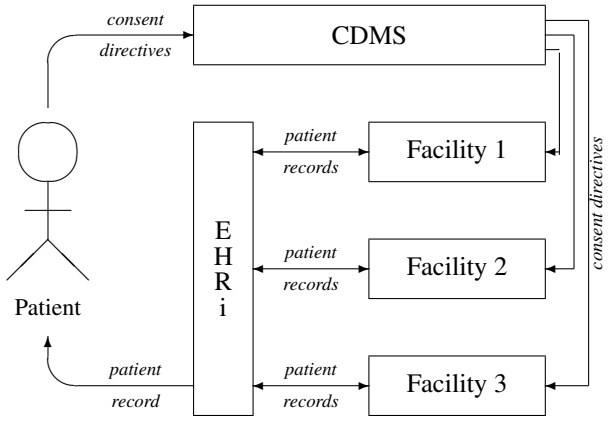

Figure 2.2: Our secure electronic healthcare record system.

- an *electronic healthcare record infrastructure*[2] ("EHRi") in which patient records are stored.

Both the consent directive management system and electronic healthcare record infrastructure are logically centralised, though they may in fact be composed of an arbitrary number of physically distributed databases.

We can think of the system shown in Figure 2.2 as being a digital rights management system in which healtcare workers are the providers, the consent directive management system is the licence issuer, the electronic healthcare infrastructure is the distributor, and the healthcare facilities are the users, with "DRM agents" implemented by the facilities' digital rights management systems (see Section 2.3 below). The implementation described in this document, however, differs somewhat from the conventional implementation of a digital rights management system in that rights-managed information is protected by means of secure authenticated channels and secure storage rather than content encryption.

Consent directives are created by patients themselves, or possibly by some legally-appointed substitute decision maker if the patient is incapable of doing so him- or herself. In general, consent directives may be initially created in the form of a paper document then converted into an electronic format by some registrar, but for our purposes we will assume that consent directives exist in an electronic form at their point of entry into the system.

Consent directives are stored in a plaintext form in the consent directive management system, where access to them is governed by an access control system. Consent directives (or the relevent portions of them) must be requested by individual healthcare facilities whenever someone in that facility makes a request to access a patient record to which that consent directive applies. Chapter 3 describes the consent directive management system in detail.

[2]Perhaps this name should be changed to distinguish it from the system as a whole?

Healthcare records are created and modified by individual healthcare workers within the facilities that are involved in the treatment of the patient to whom the records refer. Any new or modified records must be transmitted to the electronic healthcare record infrastructure, where they are stored in a plaintext form and may be later requested by other healthcare workers, or the patient him- or herself using an access control system. Chapter 4 describes the electronic healthcare record infrastructure in detail.

2.3 Healthcare Facilities

We require that all of the health-sensitive activity within a healthcare facility be controlled by *workflows* designed and maintained by that facility. Workflows set out the series of steps that must be undertaken in order to accomplish a given complex task, and are used in the proposed system to identify which people require which rights in order to carry out their work. The proposed system also uses workflows as a proxy for "purposes", so that it is possible for patients to make statements about the purpose of use of their healthcare information, and have these statements respected by considering the workflows to which their information is subjected. Chapter 5 describes workflows and their use in our system in detail.

Figure 2.3 shows an overview of a single healthcare facility, with one of its staff members. Every healthcare facility contains

- a *workflow management system* that controls all of the workflows in the organisation;
- a *licence issuer* that translates consent directives and workflow information into licences for a digital rights management system;
- a *record packager* that translates patient records into protected documents for a digital rights management system; and
- an *identity management system* that verifies credentials for indvidual workers within the facility, and assigns individual workers to roles within the facility.

Unlike the global rights management infrastructure, the digital rights management system within a facility is implemented in the conventional way, with information being protected by use of content encryption. The digital rights management system within a facility effectively implements the "DRM agent" required by the global system. Chapter 6 describes the proposed digital rights management system in detail.

All of the information flow within a healthcare facility is controlled by the digital rights management system. All information retrieved from the global electronic healthcare system (Figure 2.2) must be transformed into rights-managed information according to the digital rights management regime used at that facility, and workers within the facility must perform all tasks that relate to electronic patient records using DRM agents that conform to that regime.

All actions that require access to sensitive information are undertaken within the context of a *session*. A session is an abstract entity that is created upon the instantiation of a workflow by the workflow management system, and destroyed when that workflow

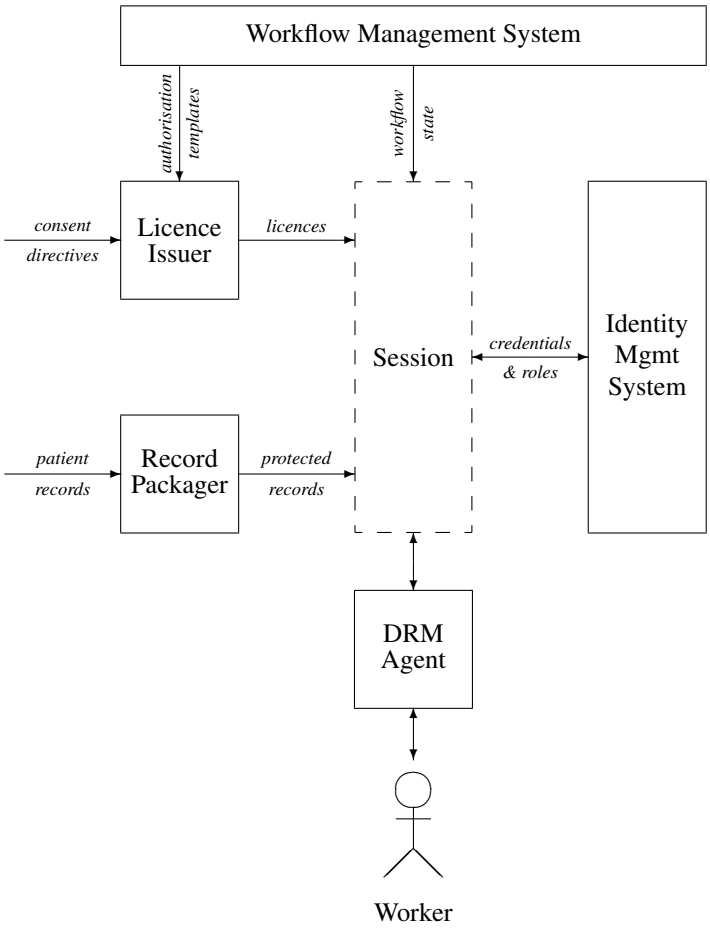

Figure 2.3: A healthcare facility and one of its workers.

concludes. All of the licences necessary to complete the workflow are issued to the session, and the worker assigned to that workflow may log in and out of the session using any suitable DRM agent.

Every state of a workflow described in the workflow management system is associated with one or more *authorisation templates* that describe the (minimum) rights required to perform the actions required by that workflow state. An authorisation template describes the rights and conditions that apply to a state, but leaves "holes" for the subject and resources that must be filled in at instantiation-time with the particular subject and resources involved in that particular workflow instance.

Upon instantiating a new workflow for a particular subject with particular resources, authorisation templates are instantiated by the licence issuer to form licences. The licence issuer must check that any licences it issues in this manner are consistent with the relevent consent directives by conferring with the global consent directive management system. Licences are issued to the session that was created for the new workflow instance.

The record packager must then retrieve the patient record to be used by the new workflow instance from the global electronic healthcare record infrastructure, and transform these into a format appropriate for the digital rights management regime in use. In this form they can be transmitted to the DRM agents used in the session.

2.4 Security

The system described in this document aims to

- minimise the amount of information that is released to users of the system, without impeding their ability to do their job; and
- ensure that any release is consistent with the consent directives of patients[3].

Chapter 9 gives a more detailed description of the security properties of the system. In broad terms, however, the security of the system relies on

- the trustworthiness and security of the global consent directive management system and electronic healthcare record infrastructure;
- the trustworthiness of the administrator of the workflow management system and identity management system in each healthcare facility;
- the trustworthiness and security of the licence issuer and record packager in each healthcare facility; and
- the security of the digital rights management system used in individual healthcare facilities.

We also require some degree of trust in the healthcare workers themselves since they will be privy to various elements of sensitive information in the course of their

[3] and, possibly, the relevent laws, but we have not yet considered these in detail.

work, and may reveal information if they are careless or malicious. (Of course, seriously malicious healthcare workers could probably do many worse things than reveal sensitive information). Our system, however, attempts to minimise the amount of information that might be leaked, and to keep workers aware of the privacy requirements of their patients.

Chapter 3

Consent Directive Management

A *consent directive* (often called a *consent object* in health informatics literature; our terminology follows [32]) is a document that describes a patient's willingness to allow information about him or her to be used. All of the consent directives in our system will be stored in a global *consent directive management system*.

We will suppose that the consent directive management system is a monolithic, globally-accessible database that stores all of the consent directives issued by all of the patients in the system. We will also suppose that the consent directive management system knows the policies of the relevent jurisdiction and is able to incorporate these policies into its decisions as necessary. (In practice, there may be one physical consent directive management system for each jurisdiction, and requests for a particular patient's consent directive must be routed to that patient's home jurisdiction. For the moment, we will not specifically consider the interaction between two different jurisdictions in the event that a patient who lives in one jurisdiction seeks treatment in another jurisdiction.)

As in other work on privacy, we will suppose that patients can give or withhold their consent to "disclose" information about them. We will assume that "disclosure" includes permission to read and modify a record as necessitated by a medical workflow, but not permission to pass the record to a third party or to make new records (such as excerpts and aggregations) based on the old one. We may consider introducing more precise actions in future versions.

We will adopt the attribute-based access control model used in the eXtensible Access Control Markup Language ("XACML") [38] for expressing consent directives and for making decisions about them. In this model, consent is expressed in terms of the values of attributes associated with the subject, action, object and environment for which a request to access information is made. Every attribute may take on an arbitrary subset of values drawn from some range for that attribute. This model generalises many other models of access control, and allows a policy to scale across a number of different organisations so long as they agree on the set of attributes and possible values to be used.

3.1 Consent Directive Creation

In general, consent directives may be created on a paper form that is filled out by a patient, or a patient's substitute decision maker in the event that the patient him- or herself is unable to fill out a form. These forms will be converted into an electronic form and stored in the consent directive management system by someone who is authorised to do so.

For our present purposes, we will not consider any paper forms, or the process by which paper forms are made into electronic documents. We will suppose that there is at least one person who is authorised to add digitised directives to the database, and that this person adds them in the form described in this chapter.

3.2 Consent Directive Expression

Consent directives will be written as a XACML rule, that is, an XML document rooted in the `Rule` element. These rules may be collected into a XACML policy (`Policy` element) or policy set (`PolicySet` element) for storage, but in this section we will focus the form of a single rule. The reader should refer to the complete XACML specification for a comprehensive understanding of XACML's syntax and semantics.

In accordance with the XACML specification, each rule (that is, consent directive) contains

- an *effect* that specifies whether the effect of this rule is to grant or deny consent;
- a *target* that specifies the subject, action, object and environment attributes to which this rule applies; and
- an optional *condition* that specifies the relationships between the attributes that must exist in order to trigger this rule.

We do not expect that conditions will be used in the initial version of our system.

A rule is triggered whenever an access request arrives with the attributes that match the target of the rule. For example, a rule might refer to all requests for object with an "object type" attribute of "X-ray". Section 3.3 below discusses the attributes and values that will be used to express targets and requests in our system.

The effect of a rule will typically be to grant consent if the relevent jurisdiction supports a "general consent" regime, and to deny consent if the jurisdiction supports a "general denial" regime. In future versions, we might also consider directives that create exceptions to broader directives by specifying a narrower target and an effect opposite to the broader directive. Such directives would allow patients to write rules for "Every surgeon except Alice", for example, by creating a directive that grants consent for all surgeons (in a general consent regime), and narrower directive that denies consent for Alice[1].

[1] This might also be done, and possibly more conveniently, using the condition of the rule.

Note. *We have used XACML here since we have adopted the attribute-based model from XACML. We could also express consent directives in an ODRL-like language in which the* `context` *element is used to hold the attributes of a party, an asset and a permission (or* `purpose` *constraint). This may have some advantages in that ODRL is somewhat less tedious and obscure than XACML for human readers, and that we have used ODRL as our rights expression language in Chapter 6. On the other hand, XACML supports all of the features we (so far) need without needing to make additions or modifications to the standard.*

3.3 Consent Attributes

In order to create and enforce an attribute-based access control policy, we need to define a standard set of attributes together with a set of values that these attributes can take on. Consent directives will be written in terms of the values of these attributes, and decisions to disclose information will be made according to the value of the attributes possessed by the entities involved in the disclosure.

In this section, we follow draft requirements provided by Canada Health Infoway [32] to develop a set of attributes necessary for expressing the kinds of consent directives contemplated by the authors of the draft. Other attributes may be useful in other contexts.

Subject Attributes. Following [32], every healthcare worker will be associated with an attribute for each of

- that worker's unique identifier;
- the roles to which that worker belongs;
- the facility in which that worker is employed; and
- any "circles of care" to which that worker belongs (see below).

Each worker may belong to an arbitrary number of roles, and may belong to an arbitrary number of circles of care at any one time. Each worker, however, has only one unique identifier and works for only one facility.

The concept of a "circle of care" does not currently have any formal definition. For our purposes, however, we will define the circle of care of a patient to be the set of workers engaged in a workflow that relates to that patient. Thus the members of a circle of care can be identified in an automated fashion by looking up workflow information. We will discuss workflows in detail in Chapter 5.

We will suppose that there exists some globally-recognised set of possible values that each of the foregoing attributes can take on. Individual facilities may use their own methods of identifying workers, roles, and so on if they wish, but any local identifiers must be translated into their global equivalents when communicating with the global consent directive management system.

In our system, the identity management system of each facility will be responsible for storing the mapping between the workers in that facility, and the values of their

attributes. A worker's attributes may be retrieved from the identity management system in the form of an assertion in the Security Assertion Markup Language [39], as described in Chapter 7. We assume that the identity management system of a facility is trusted to associate workers in that facility with the correct attribute values.

Object Attributes. Also following [32], every electronic healthcare record will be associated with an attribute for each of

- the unique identifier for that record;
- the patient to which that record refers;
- the health domain to which that record applies; and
- the episode of care to which that record refers.

Each of these attributes will have only one value[2].

As for subject attributes, we will suppose that there exists some globally-recognised set of possible values (identifies) for each attribute above. Since records are stored in the global electronic healthcare record infrastructure, there should be no reason for facilities to use their own local identifiers, with the possible exception of identifiers for episodes of care.

In our system, the attributes of all records will be stored in the global electronic healthcare record infrastructure alongside the records themselves. We will suppose that they are stored in the same database as the records themselves, and that they can be retrieved using the same interface as the one used for retrieving the records.

Action Attributes. Since there is only one action ("disclose") in our system, actions do not need explicit identifiers of the sort that subjects and objects require. We assume that facilities do not grant any rights that are not covered by the "disclose" action. Licences created within a healthcare facility may contain distinct actions such as "read", "modify", etc., but these will all be translated into "disclose" for the purposes of obtaining consent.

In addition to the actions themselves, much work in privacy suggests that consent directives be associated with a "purpose"[3]. XACML's Privacy Protection Profile proposes that this concept be represented as an action attribute, and we will follow this convention here. Thus every action will be associated with an attribute whose value identifies the purpose of this action.

We will suppose that there exists some globally-recognised set of values that refer to a well-known set of purposes. Every workflow stored in a facility's workflow management system (see Chapter 5) will be associated with one of these purposes, and any action undertaken as part of that workflow will be associated with an attribute that identifies the purpose of the workflow. We will assume that the policy officer an organisation is trusted to associate the workflows that he or she designed with an appropriate purpose.

[2]Unless it is possible for a record to refer to more than one domain?

[3]Canada Health Infoway's draft has consent directives referring to an "indication", which we have taken to mean a purpose in the sense of other work in privacy.

3.4 Consent Requests

In our system, healthcare workers do not directly request permission to access a healthcare record from the global consent directive management system. Instead, the licence issuer of the facility constructs a licence that permits access according to a workflow, as described in Chapter 5. The licence issuer must then request permission to issue the licence from the global consent directive management system by transforming this licence into an access control request.

As in XACML, a request for consent takes the form of

- a list of the attributes possessed by the beneficiary of the licence;
- a list of the attributes possessed by the resource of the licence;
- a list of the attributes of the permission (that is, the `purpose` constraint in the ODRL licences used in Chapter 6) of the licence; and
- a list of the attributes possessed by the environment to which the licence is restricted (that is, temporal, spatial and other miscellaneous constraints of the licence).

We will not use environment attributes in the initial version of our system.

Upon receiving a request from a licence issuer, the consent directive management system will compare the request with the consent directives using the usual XACML decision algorithm, and return the response to the licence issuer. If the response is positive, the licence issuer may issue the licence. Otherwise, it must not (and presumably report an error to the person who requested the licence).

3.5 Obligations

In general, it is possible for XACML rules to contain obligations (identified by some unique identifier) that subjects must undertake after being granted access to some object. The identifiers of these obligations are returned with the access control decision. In our system, such obligations could either be carried out by the licence issuer itself, or passed on to individual healthcare workers by incorporating them as a `requirement` element in an ODRL licence.

Obligations – notably obligations to log all accesses – seem useful in the healthcare scenario, and we may consider adding support for them in a future version of our system.

Chapter 4

Electronic Healthcare Record Infrastructure

For our purposes, we will treat the electronic healthcare record infrastructure as a monolithic, globally-accessible database that stores all of the patient records in the system in their plaintext forms. In practice, the infrastructure may actually be a federation of physically-separated databases, with some mechanism to route requests for particular healthcare records to the member database in which the desired record is stored.

We will suppose that every patient is associated with an arbitrary number of atomic healthcare records that result from different encounters with the healthcare system. Every record is required to have a unique identifier, and we suppose that healthcare records can be retrieved either by use of this identifier, or by the identifier of the patient to whom the record refers. (In our examples, we will use healthcare record identifiers that incorporate the patient's identifier). The format of the healthcare records themselves is beyond the scope of the present document[1].

Access to the infrastructure will be governed by a conventional access control system so that only the digital rights management systems of participating healthcare facilities are permitted to access the database. We will assume that all such access is conducted over a secure authenticated channel, and that the digital rights management systems are trusted to read, write and create records (and their attributes, as described in Chapter 3) in the database without further supervision.

In a future version of the system, it may also be possible for patients to access their own healthcare records by connecting directly to the electronic healthcare infrastructure via some sort of Internet "healthcare portal". This requires the access control system to be somewhat more complicated, since it then needs to authenticate arbitrary users and it may need to mask certain portions of a healthcare record if these have been blocked by a physician.

[1] We may re-consider this when we consider records with partial masking.

Chapter 5

Workflows

A *workflow management system* automates the business processes of an organisation that require members of the organisation to carry out a set of tasks according to some rules in order to achieve some complex end. In this document, we will follow *Yet Another Workflow Language* ("YAWL"), developed by van der Aalst and ter Hofstede after a broad survey of existing workflow management systems [51, 58]. Other models could be used without signficantly affecting our system, and, indeed, different facilities may use different workflow management systems in practice.

Aside from their usual functions in managing the processes of a healthcare facility, workflows are used in our system to

- identify the "circle of care" of a patient as the set of healthcare workers involved in the workflows created to treat that patient; and
- define an authorisation policy that allows healthcare workers to access sensitive information as required by their work, but not otherwise.

Individual workers may join and leave the circle of care as new members are employed to carry out new sub-workflows, or old sub-workflows are completed and the worker responsible moves on to another task.

Numerous authors have proposed access control models in which access to resources is controlled by workflows [2, 8, 9, 11, 17, 22, 24, 27, 29, 30, 42, 45, 54, 55, 56, 57]. Our treatment adopts the notion of an *authorisation template* introduced in the fundamental work of Atluri and Huang [2], which we use as a template for generating digital rights management licences corresponding to each state of a workflow.

We will assume that workflows, and the authorisation policies that they imply, are designed by some trustworthy *policy officer*. The policy officer is trusted to design workflows that accurately represent the work that needs to be done to achieve the organisation's goals, and to specify the minimum level of rights required to complete a workflow successfully. The reader is directed to work in adaptive workflows for more sophisticated models of access control for workflow descriptions [7, 16, 41, 52].

5.1 Workflow Model

A workflow in YAWL takes the form of a directed graph in which each node is either a *condition* or a *task*. Arcs may flow from conditions to tasks, tasks to conditions, or tasks to tasks, but not from conditions to conditions. The workflow starts at a unique *input condition* and must conclude at a unique *output condition* (that is, there is only one possible way in which a workflow can be started, and only one possible way in which it can conclude).

Following Russello, et al. [45], we will assume that any one workflow refers to the actions of only one person, called the *subject* of the workflow. We will refer to such workflows as being *atomic*. Complex workflows that involve interactions between multiple people must by decomposed into atomic workflows that refer to the individual workflow of each person. A single atomic workflow may, however, refer to an arbitrary number of (components of) patient records and other objects, which we will refer to as being the *resources* of the workflow.

While it is possible to identify a specific subject and resource in a workflow, we expect that it would be more usual for the subject and resources of a workflow to be different every time the workflow is executed. Thus a workflow's subject and resources would be specified as place-holders of various classes at design-time, with the intention that these place-holders be replaced by an actual member of the class when the workflow is executed.

The workflow is executed by a set of *tokens* that move from one node to another. There is initially only one token (at the start node), but new tokens may be created by *splits* in which the completion of one task makes several new tasks possible. Splits are represented in a workflow graph as a task with an out-degree of greater than one.

YAWL has three kinds of splits:

- *AND-splits* in which the completion of one task requires all of the following tasks to be executed;
- *OR-splits* in which the completion of one task requires at least one of the following tasks to be executed; and
- *XOR-splits* in which the completion of one task requires exactly one of the following tasks to be executed.

AND- and OR-splits result in new tokens being created in the workflow, one for each task that is to be completed following the task that caused the split.

Tokens can be removed by an analogous *join* operation, represented as a node with an in-degree of greater than one. There are three kinds of joins, analogous to the three kinds of splits:

- *AND-joins* require all of the preceding tasks to be completed before the joined task can be started;
- *OR-joins* require at least one of the preceding tasks to be completed before the joined task can be started; and

- *XOR-joins* require exactly one of the preceding tasks to be completed before the joined task is started.

Tasks can also cancel other parts of a workflow, which causes all of the tokens in the cancelled part to be eliminated.

5.2 Workflow Execution

A particular workflow may be initiated by a person to whom the *execute* right for that workflow has been granted by the digital rights management system, as described in Chapter 6. The initiator of a workflow must choose a subject who will carry out the workflow, and must associate the workflow with a particular set of resources that will be used in the execution of this instance of the workflow. The choice of subjects and resources may be constrained by typing information stored in the workflow description, or by constraints in the licence from which the *execute* right was derived.

Initiating a workflow will cause the workflow management system to

- create a new instance of that workflow in its memory space;
- position a token in the workflow's input condition; and
- create a new authorised session (see Chapter 6) for the subject of the workflow.

Tokens will move through the workflow instance according to the tasks completed in the session, as described in Section 5.1 above. When all of the tokens have reached the output condition or been cancelled, the workflow instance and the session will be destroyed.

At any one time in our system, a session will be associated with a set of *current* tokens that represent the set of tasks that must be done, and a set of *potential* tokens that represent the set of tasks that the owner of the session may choose to pursue (but does not have to). Initially, there is only one current token and no potential tokens.

At any time, a DRM agent in the session may choose one of the current or potential tokens and perform the task that it represents. Upon completing a task, we will require the DRM agent responsible to authenticate itself to the workflow management system and request that the token in that task be advanced:

- if there is no split in the workflow, the token will advance to its next state and be added to the set of current tokens;
- if there is a XOR-split in the workflow, the DRM agent will be asked to choose the next state, and the token will be advanced to this state and added to the set of current tokens;
- if there is an AND-split in the workflow, current tokens will be created for all of the following states; and
- if there is an OR-split in the workflow, potential tokens will be created for all of the following states.

If the "next state" is the end condition, the token will disappear.

5.3 Services and Cross-Organisational Workflows

Our model for cross-organisational workflows is similar to that of Kang, et al. [24], who view a cross-organisational workflow as being a composition of atomic workflows, each controlled by a single organisation. To this end, we will require that each facility exposes a set of workflows tha can be initiated by entities outside of that facility. We will refer to these workflows as being the *services* of the facility.

In this model, complex workflows may cross organisational boundaries by decomposing them into a set of atomic workflows, such that each atomic workflow is "owned" by a single facility but may be exposed as a service to other facilities. We will not consider how cross-organisational workflows are designed, but will suppose that the policy officers of the participating facilities somehow come to an agreement by which they design their individual workflows to meet the needs of the cross-organisational workflow.

Each service workflow, whether it is part of a cross-organisational workflow or not, will be managed by the workflow management system of the facility that exposes the service, and will be executed by some member of that facility. Service workflows, therefore, are the same as other atomic workflows within the facility, except that they are initiated by entities outside the facility.

While a facility may have a number of services that can be invoked by anyone (these might represent services that it offers to the public, for example), in general we expect that a facility will only allow certain people or kinds of people invoke a particular service. Since these people are from outside the facility that owns the workflow, this cannot be done through the facility's digital rights management system as for internal workflows.

We will suppose that a facility that offers a service as part of a cross-organisational workflow has an agreement with any facility whose members might call that service, such that the workflow management system of the service provider will recognise an *execute* right issued by the licence issuer of the service invoker, when it relates to a service exposed to that invoker.

5.4 Authorisation Templates

Following Atluri and Huang [2] and Russello, et al. [45], we will require the policy officer of a facility to associate every task of every workflow with an *authorisation template* that describes the minimum rights required to accomplish that task.

An authorisation template is an authorisation policy (in our system, a licence) with one or more *holes* that must be filled by the initiator of a workflow that uses this template. Holes may be "typed", so that they can only be filled by entities of a particular type, that is, role in the case of a subject and class in the case of a resource.

In our model, all authorisation templates will have at most one *subject hole* that will be filled with the subject of the workflow instance (there will be no subject hole only in the case that some workflow can only be performed by one identified individual). Each authorisation template may also have an abritrary number of *resource holes* that will be filled in with the resources of the workflow instance.

We will refer to the process of creating an ordinary licence from an authorisation template as *instantiation*. The licence issuer of a facility my instantiate an authorisation template by replacing its holes with appropriate subjects and resources, thus forming an ordinary licence written in the rights expression language of the digital rights management system. This licence may then be issued as usual.

We will discuss the format of authorisation templates in detail in Chapter 6.

Chapter 6

Digital Rights Management

The digital rights management system of a facility provides the basic mechanism by which security is enforced within that facility. In principle, it is possible for each facility to install its own digital rights management system that employs its own particular encryption methods, rights expression language, etc., so long as the system meets the security and inter-operability requirements of the global secure electronic healthcare infrastructure. For our purposes, however, we will assume that all facilities implement the digital rights management system described in this chapter.

We will make all of the usual assumptions for a digital rights management system, namely that

- all DRM agents are trusted to comply by the rules set out in licences issued to them, and that DRM agents can be authenticated using some public key infrastructure;
- the licence issuer of a facility is trusted only to issue licences in accordance with the policy of the information owners (here represented by the consent directive management system and the workflow management system);
- the record packager of a facility is trusted to distribute information in a protected form only, as dictated by the licence issuer; and
- any domain or session controllers (see Section 6.1 below) are trusted to admit devices to domains or sessions only in accordance with some policy established by the information owners.

We will assume that the licence issuer, record packager, workflow management system and identity management system of each facility has some well-known public key by which the DRM agents within that facility can authenticate them. These keys would presumably be distributed during some set-up phase.

6.1 Authorised Sessions

We will introduce the notion of an *authorised session* (in the model of an *authorised domain*) in our system. We use authorised sessions to capture the notion of a complex rights-managed task (here, a workflow) being carried out by the use of a number of distinct DRM agents who all share the state of the task. Authorised sessions allow licences to be issued for the session as whole rather than to the individual DRM agents involved in the task.

An authorised session is akin to an authorised domain and supports similar membership operations, but differs in that all of the state information associated with a session is shared by all of the members of session. That is, all constraints are "shared" in the sense of [48]. In the system being described in this document, there will usually only be one member of a session at any one time and we are using the notion of a session as a means of transferring the state of a workflow from one device to another. In general, however, a session may have several concurrent members who all participate in the session simultaneously using some common channel.

Every authorised session has a unique identifier by which it can be identified in a licence. To ensure that there is no ambiguity in rights expressions that refer to sessions and domains, we will suppose that the identifiers of sessions and domains are disjoint, that is, that there are no identifiers that could refer to both a session and a domain.

Sessions have a *session controller* analogous to the domain controller of an authorised domain, and the session controller will support some "join session" and "leave session" protocols analogous to the equivalent protocols in an authorised domain system. For our purposes, these protocols and the cryptographic scheme that they support, will be identical to the protocols and cryptographic scheme used by authorised domains of the same digital rights management system. (In a primitive implementation, an authorised session may actually just be an authorised domain.)

6.1.1 Session Lifecycle

In our system, every workflow instance will be associated with a unique session that acts as the subject of the workflow. This session will be created by the workflow management system[1] as part of initiating a workflow. The session will be similarly destroyed by the workflow management system when the workflow instance is completed or cancelled.

6.1.2 Session Membership

In general, membership of a session will be controlled according to some *session policy*, as for the "domain licensing" model that we proposed for authorised domains in [48]. For our present purposes, we will only use membership criteria that refer to user roles, including "unit roles" whose only member is a specific human user. The assignment of users to roles is stored by the identity management system.

[1] or, possibly, requested to be created, with the actual creation being done by some stand-alone session controller.

In order to join a session, a DRM agent must prove that (a) it is a trusted DRM agent and (b) its current human user is a member of the role or roles for which that session was created. It will do the former using the usual authentication method for the digital rights management system, and the second by supplying a DRM user assertion as described in Chapter 7.

A DRM agent may leave a session by initiating the "leave session" protocol of the digital rights management system with the session controller. We can see no reason for a (well-formed) "leave session" request to be refused.

6.2 Rights Expression Language

Licences in our system will be written in the Open Digital Rights Language ("ODRL") [36]. Similar licences could be written in the the Extensible Rights Markup Language ("XrML") [13], but we have chosen ODRL due to its greater accessibility.

Unless otherwise described in this chapter, all ODRL elements will have their usual syntax and semantics as described in Version 1.1 of the ODRL specification[2]. This chapter should be read in conjunction with the full specification for a complete understanding of the licences used in our system.

6.2.1 ODRL Templates

We will express authorisation templates using an extended form of ODRL, and we will refer to documents written in this language as *ODRL templates*.

An ODRL template may contain any of the elements of the ODRL Expression Language (prefix `o-ex`) and ODRL Data Dictionary (prefix `o-dd`) namespaces, together with several new elements from an ODRL Template namespace (prefix `o-t` in this document) that are described in this section.

Root Element. The root element of an ODRL template will be a `template` element. In an ODRL agreement generated by instantiating this template, this element will be replaced by the `rights` element of the ODRL Expression namespace.

Holes. "Holes" will be expressed in a template using the `forany` element. The `forany` element may occur as the child of any `party` or `asset` element. In an ODRL agreement generated by instantiating this template, the `forany` element will be replaced by a `context` element that specifies a concrete party or asset, as appropriate.

If the `forany` element has a `context` child, the corresponding party or asset in the instantiated agreement must be drawn from the template's context. For example, if the template specifies a role for a party, the instantiated agreement must be made with a member of that role.

The `forany` element may have an attribute `var` that associates a variable name (an arbitrary string) with a hole. If two `forany` elements in the same template have the

[2]A working draft of Version 2.0 of ODRL exists at the time of writing. Version 2.0 may have some advantages compared to Version 1.1 but we are following the current standard for now.

same variable name, they must be instantiated with the same context in an agreement generated by instantiating this template.

Note. Our `forany` element is modelled on the `ForAll` element of XrML, and XrML templates might be written using the latter element. There is a subtle difference in the semantics of each element, however, as the `ForAll` element is used to write a licence that refers to *all* of the members of a class, while the `forany` element is used to write a template from which a licence can be generated that refers to *any* member of the class (but will not necessarily be issued for any particular member.)

6.2.2 Rights and Workflows

Identifying Workflows in Licences. All workflows stored in the workflow management system will have a unique identifier, which can be used as the asset of an ODRL agreement using the usual syntax for referring to resources with a unique identifer, that is, by use of the `uid` context element.

We will use the existing `execute` permission of ODRL to represent the act of initiating a workflow, that is, a licence with a workflow as its asset and `execute` as its permission permits its beneficiary to initiate that workflow. The `execute` permission has its usual meaning if the licence's asset is a binary executable (though this is unlikely to be of much interest in our system).

Indirect Assets. A workflow may make use of an arbitrary number of resources (assets in ODRL), which can be specified as children of the `permission` element of an initiation licence. We will refer to such assets as being the *indirect assets* of the licence (as opposed to the *direct asset*, which is the workflow). If indirect assets are specified by the licence, the workflow may only be executed using those assets as its resources.

If the workflow uses more than one resource, we will require that each of these resources be identified by some parameter name (in the sense of a function call) in the workflow description. Every indirect `asset` element in an initiation licence for this workflow must contain a `param` attribute that specifies the parameter to be filled by this asset.

Workflow Purposes. We will use the existing `purpose` constraint of ODRL to constrain the exercise of a permission to a particular state of the workflow, that is, a permission subject to a `purpose` constraint may only be exercised when the beneficiary (a session, in our system) is at the state of a workflow identified by the context of the `purpose` element.

We will require that every workflow state has a unique identifier that can appear as the `uid` context of a `purpose` element. We will also allow `purpose` constraints to refer to workflows as a whole using their unique identifiers, so that it is possible to write licences that are valid (only) for a beneficiary who is engaged in some state in that workflow.

6.2.3 Creating Records

In addition to modifying existing records according to the `modify` and `annotate` permissions of ODRL, healthcare workers may also create entirely new records that refer to new episodes of care, new patients, and so on. Our rights expression language will include a permission called `create` that refers to this operation.

A licence that awards the `create` permission may specify the attributes (unique identifier, type, etc.) to be associated with the new record by including them within the `context` child of the `asset` element. If an attribute is not fixed by the asset's `context` child, the agreement's beneficiary may choose an arbitrary value for this attribute[3].

6.3 Cryptographic Architecture

We will adopt the cryptographic architecture of the Open Mobile Alliance's Digital Rights Management Specification Version 2.1 ("OMA DRM") [37], with some slight simplifications. This architecture is much the same as the one that we previously used in developing the SITDRM Enterprise system [47].

As in SITDRM, we will use

- the RSA algorithm for all asymmetric-key cryptographic operations;
- the AES algorithm for all symmetric-key cryptographic operations; and
- the SHA-1 algorithm for all cryptographic hashes.

6.3.1 DRM Agents

We will assume that every trusted DRM agent possesses a unique asymmetric key pair, of which the private key is known only to the DRM agent. We will assume that a DRM agent's public key can be verified using some public key infrastructure that is not part of the present proposal. As in OMA DRM and SITDRM, a DRM agent's public key will act as its unique identifier[4].

We will assume that every DRM agent contains a body of secure storage in which it may store an arbitrary amount of cryptographic information transmitted to it by other actors in the system. We will assume that this storage cannot be read, modified or replayed by an attacker.

6.3.2 Licence Issuer and Record Packager

We will assume that the licence issuer and record packager of every facility each have a well-known public key and corresponding private key known only to themselves. We

[3] We suppose that healthcare workers will choose attributes according to some naming scheme not described in this document.

[4] OMA DRM actually uses the hash of the public key; this may make writing licences slightly more convenient.

will assume that every DRM agent within that facility is able to verify these public keys using some public key infrastructure.

6.3.3 Authorised Domains and Sessions

Every authorised domain and session will be associated with a unique symmetric key called the *domain key* or *session key*, as in OMA DRM. Upon joining a domain or session, a DRM agent will receive a copy of the domain or session key, and store it in its secure storage. Upon leaving a domain or session, a DRM agent will delete the corresponding key.

As for the licence issuer and record packager, we will assume that the domain controller and session controller (that is, workflow management system) of a facility each have a well-known public key and corresponding private key known only to themselves. We will assume that every DRM agent within that facility is able to verify these public keys.

6.3.4 Record Encryption Keys

Every protected record (that is, the output of a record packager) will be encrypted by a randomly-chosen symmetric key called the *record encryption key*, as in OMA DRM. The key will initially be known only to the record packager that created the protected record, and the licence issuer of the same facility.

Two different record packagers may (and probably will) choose different record encryption keys for the same record. Thus protected records cannot, in general, be moved directly from one facility to another. (A user in a facility to which a protected record has been referred must re-request the record from his or her own facility's record packager.) We may re-visit this in a later version of the system.

6.3.5 Licences

Every licence that grants a permission over a protected record will contain the record encryption key for that item, encrypted by

- the public key of the DRM agent, if the beneficiary of a licence is an individual DRM agent; or
- the domain or session key, if the beneficiary of a licence is an authorised domain or session.

The encrypted key will be stored as an XML Security `KeyInfo` element within the `asset` element of a licence.

Note that licences that grant permissions over workflows will not contain any key information, since workflows are not encrypted. All licences that refer to healthcare records, however, will contain keys as records must be distributed in an encrypted form.

All licences will be signed by the licence issuer that issued them. The signature will be contained within an XML Security `Signature` element within the root element of a licence.

Note. This algorithm is essentially the same as that used in SITDRM, except that domain and session keys are symmetric here. The format of licences is the same as that used in the unpublished "Corrimal" licence transfer demonstrator developed at the University of Wollongong. This algorithm is slightly different from that used in OMA DRM, in which each licence is encrypted by a "rights encryption key" and it is this key that is encrypted by the DRM agent or domain key.

Chapter 7

Identity Management

The identity management system of a healthcare facility stores

- credentials for the human users within the facility; and
- assignments of human user to roles within the facility.

Credentials may take the form of passwords, secrets for authentication tokens, or any other authentication mechanism deemed suitable by the healthcare facility. We will suppose that these credentials and role assignments are created by the facility's policy officer during some set-up phase.

The main function of the identity management system will be to provide security assertions in the Security Assertion Markup Language ("SAML") [39] that allow other parts of the system to verify the identities and roles of human users. There are two main kinds of assertion required by our system:

- a *DRM user assertion* that asserts that a particular DRM agent is being used by a particular human user; and
- a *user role assertion* that asserts that a particular human user is a member of a particular role.

This chapter describes the contents of the *attribute statement* of an assertion, which is used to assert the values of particular attributes for a given subject. The reader should refer to the SAML specification for a complete understanding of the syntax and semantics for assertions.

7.1 DRM User Assertions

A *DRM user assertion* asserts that a particular DRM agent is being used by a particular human user, and that that human user is acting in a particular set of roles. The subject of a DRM user assertion will be a DRM agent, and the assertion will contain

- one `AttributeStatement` element that identifies the current human user of the DRM agent; and

- one `AttributeStatement` element that identifies all of the roles that have been activated by that human user.

7.2 User Role Assertions

A *user role assertion* asserts that a particular human user is a member of some set of roles. The subject of a user role assertion will be a human user, and the assertion will contain a single `AttributeStatement` element that lists all of the roles to which that user belongs.

Chapter 8

Processes

In this chapter, we will describe the technical steps taken by various actors in system in order to carry out particular tasks. A real system would automate most, if not all, of these steps beyond the initial step triggered by a human user who wanted the task to be performed. An example of how these technical processes fit together in a whole workflow is given in Appendix A.

8.1 Workflows

All of the ordinary actions undertaken within a facility are governed by workflows, as described in Chapter 5.

Some roles or individuals must be given permission to initiate certain workflows that we will call the *root workflows*. These workflows may contain states that require the initiation of other workflows, which themselves may contain states that require the initiation of further workflows, and so on.

Licences for the initiators of root workflows must be issued directly to those initiators during some set-up phase. We will suppose that the facilty's licence issuer will issue these licences at the request of the facility's policy officer, and that the policy officer will transmit them to the initiators as necessary. Since these licences to not refer to health records, we do not require them to be vetted by the global consent directive management system as for other licences.

Licences for the initiators of other workflows are created from the authorisation template associated with the workflow state that requires the workflow to be initiated. We will discuss instantiation of authorisation templates in detail shortly.

8.1.1 Workflow Initiation

A DRM agent in possession of a valid `execute` permission for a workflow (whether it receives this permission directly, or via an authorised domain or session) may apply to the workflow management system to instantiate the workflow identified as the asset of the licence:

1. The DRM agent authenticates itself to the workflow management system and transmits the initiation licence (or relevent information from it)[1].

2. The workflow management system looks up the workflow description and asks the DRM agent to supply the subject of the new workflow instance and the identities of resources to be used by the workflow.

3. If these are not enforced by the licence, the DRM agent asks its user to specify them.

4. The workflow management system checks that the proposed subject is a member of the role for which the workflow is intended, as described in Section 8.3.2 below.

5. *The workflow management system should also check that the proposed resources are of the correct kind, but we do not yet have a protocol for this.*

6. The workflow management system creates a new instance of the workflow to be initiated, including an authorised session within which the new workflow will be executed.

7. The workflow management system asks the licence issuer to instantiate all of the authorisation templates for the new workflow instance (see Section 8.1.3 below).

8. The workflow management system transmits the new session identifier to the initiating DRM agent.

9. The initiating DRM agent transmits the session identifier to the human user who will be the human subject of the new workflow instance.

10. The human subject of the new workflow instance instructs his or her DRM agent(s) to join the session for this workflow.

11. The workflow management system transmits all of the instantiated licences for this session to the DRM agent(s).

12. The DRM agent(s) begin executing the workflow as described in Chapter 5.

There are a number of possible variations on the order of the steps given above, and in which components are responsible for transmitting and storing licences and session identifiers.

8.1.2 Service Invocation

Invoking a service is similar to initiating a workflow, except that the initiator is an entity from outside the facility. We will suppose that requests to invoke a service are handled directly by the workflow management system, though in a real system there may be some intermediary between outsiders and this system.

[1]The workflow management system could also verify that the licence is valid, but for now we will suppose it trusts the DRM agent to do this

We will suppose that the workflow management system of a facility recognises licences issued by the licence issuer of any facility with whom the workflow management system's home facility has a suitable agreement. Upon receiving a licence from a DRM agent seeking to invoke a workflow, it must

1. verify that this licence was issued by a licence issuer with which it has an agreement;

2. verify that this licence complies with the same agreement; and

3. verify that the invoking device is, in fact, a legitimate beneficiary of that licence.

For our purposes, we will assume that the workflow management system accepts any licence so long as the licence is signed by the licence issuer of a facility with whom it has an agreement.

To verify that the invoking DRM agent is a legitimate beneficiary of the licence, the workflow management system must verify that it is a member of the session to which the licence is issued. This can be done by querying the session controller (that is, workflow management system of the invoking organisation) over an authenticated channel[2].

Once verification has succeeded, the workflow management system initiates the workflow that corresponds to the service as described in Section 8.1.1 above, starting from Step 2. We will suppose that the invoking DRM agent chooses the subject of the new workflow as for an internal workflow. In a real system, the subject would more likely be appointed by some entity within the facility offering the service.

8.1.3 Authorisation Template Instantiation

The licence issuer of a facility accepts requests to instantiate authorisation templates from the workflow management system of the same facility (only) as follows:

1. The workflow management system establishes a secure authenticated channel with the licence issuer and transmits the identifier of the session for which it wants authorisation templates to be instantiated.

2. The workflow management transmits the first authorisation template to be instantiated, together with the identifiers of the resources with which it is to be instantiated.

3. The licence issuer constructs a prospective licence from the authorisation template, the resource identifiers, and the session identifier.

4. The licence issuer obtains the attributes of the subject of the licence from the identity management system.

[2]Alternatively, we could have the DRM agent obtain a ticket from its home session controller prior to invoking the service, and present this ticket to the foreign workflow management system as part of the invocation request.

5. The licence issuer obtains the attributes of the resource of the licence from the global electronic healthcare record infrastructure[3].

6. The licence issuer obtains the attributes of the action from the workflow management system.

7. The licence issuer makes an access control request for these attributes to the global consent directive management system.

8. The consent directive management system compares the request with its policy and returns a response to the licence issuer, as described in Chapter 3.

9. If the response is negative, the licence issuer aborts the process and reports an error.

10. Otherwise, the licence issuer obtains the record encryption key for any resources in this licence from the record packager of the same facility, using a secure authenticated channel.

11. The licence issuer encrypts the record encryption keys with the session key and inserts them into the licence.

12. The licence issuer signs the licence and transmits it to the workflow management system.

13. The workflow management system transmits the second authorisation template to be instantiated, then the third, and so on, until all of the templates templates have been instantiated, or the process has been aborted due to lack of consent.

Again, there are several variations on the order in which the above steps could be taken.

8.1.4 Purpose Constraints

When a DRM agent is asked to exercise a permission that is subject to a `purpose` constraint of the kind described in Chapter 6, it must verify that the workflow it is working on is, in fact, in the state demanded by the constraint. That is, it needs to check that the session has a current token in the state identified by the `purpose` constraint.

An on-line DRM agent could do this by authenticating itself to the workflow management system and querying the position of the current tokens in the workflow associated with the session to which the licence has been awarded. It could similarly query for higher level purposes, such as the purpose of the workflow as a whole.

We will assume, however, that a DRM agent can store the current state of any workflow instances to which it belongs in on-board secure storage, in which location it can be accessed even when the DRM agent is off-line. The DRM agent must be on-line in order to advance the workflow state, but thereafter may be off-line for an arbitrarily long period of time before it needs to advance the workflow to the next state.

[3]We would also have it that the consent directive management system retrieves these itself.

8.2 Digital Rights Management

In keeping with the cryptographic architecture of our digital rights management system, we will adopt the "Rights Object Acquisition Protocol (ROAP) Suite" from the Open Mobile Alliance's Digital Rights Management Specification Version 2.1 [37]. We will not use all of the protocols specified in the ROAP Suite, but only those noted below. The complete descriptions of these protocols can be found in the specification documents.

The ROAP protocols contain their own security mechanisms and do not need to be executed over a secure authenticated channel. (Indeed, this would not be possible for the registration protocol, since at this point the DRM agent and the licence issuer do not, in general, recognise each other). We will assume that confidentiality of the messages is not important except insofar as any sensitive cryptographic information (domain keys, record encryption keys, etc.) must not be leaked outside of the digital rights management system. The identities of devices, the membership of domains and sessions, and so on, may be public.

8.2.1 Registration

The registration protocol is used to establish a relationship between a DRM agent and a licence issuer, or domain or session controller (in OMA DRM these are all the same entity). In our system, it would be executed when a new device is brought into a facility so that the new device can learn the public keys and other parameters used in that facility. The protocol proceeds as follows, where we use "server" as a generic term for the licence issuer, domain controller or session controller:

1. The DRM agent sends its identifier to the server.
2. The server sends its identifier to the DRM agent.
3. The DRM agent sends its public key certificate to the server.
4. The server checks that the DRM agents' certificate is valid (using some public key infrastructure and the On-Line Certificate Status Protocol [33]).
5. The server sends its certificate to the DRM agent.

We may omit the registration process from the initial version of the system, and simply assume that all of the entities involved are already acquainted with one and other.

8.2.2 Licence Acquisition

The licence acquistion protocol (called the "rights object acquisition protocol" in OMA DRM) is used by DRM agents to obtain licences from the licence issuer or session controller. In general, the protocol may be initiated by the DRM agent using a request message, or the server may simply send a "response" without any prompting. In our system, most or all licences will be acquired from the session controller that is responsible for executing a particular workflow as follows:

1. A DRM agent joins a session by executing the Join Domain Protocol described in Section 8.2.3 below.

2. The session controller sends a licence acquisition response that contains all of the licences for that session.

8.2.3 Joining Domains and Sessions

The protocols for joining an authorised domain and an authorised session are identical, and we will suppose that the domain or session controller can determine which is intended by the identifier of the domain or session. The protocol proceeds as follows:

1. The DRM agent sends the identifier of the domain or session that it wants to join to the controller.

2. The controller verifies that this DRM agent is permitted to join this domain or session, according to the policy of the domain or session (see below).

3. If successful, the controller sends the domain or session key to the DRM agent, encrypted by the public key of that DRM agent.

In our system, a device may join a workflow session if and only if its current human user is the subject of that workflow. We will suppose that the session controller is able to verify a DRM user assertion of the kind described in Chapter 7. Prior to joining the session, the user must instruct his or her DRM agent to obtain an appropriate DRM user assertion from the identity management system, as described in Section 8.3.1 below. This assertion must then be included in the request to join the session, and the session controller must verify it before allowing the DRM agent to join the session. We will assume that DRM agents automatically leave all sessions when their user logs out.

8.2.4 Leaving Domains and Sessions

A DRM agent may leave an authorised domain or session by sending a leave request to the domain or session controller. The request must contain the identifier of the domain or session, and must be signed by the DRM agent. We will suppose that the controller never refuses such requests, except when the signature cannot be verified. The controller will send a response to the DRM agent that confirms its departure from the domain.

8.3 Identity Management

8.3.1 Acquiring DRM User Assertions

We will suppose that DRM agents are to automatically acquire an appropriate DRM user assertion every time a user logs in, as follows:

1. The DRM agent establishes a secure authenticated channel to the identity management system.

2. The DRM agent obtains credentials from its user and transmits them to the identity management system.

3. The identity management system checks the user's credentials, and returns a response to the DRM agent.

4. If the credentials are accepted, the user chooses which roles he or she wishes to activate (these may also be chosen automatically), and the DRM agent transmits these to the identity management system.

5. If this set of roles is acceptable, the identity management management system creates a DRM user assertion as described in Chapter 7 and transmits this to the DRM agent.

We will assume that the DRM agent automatically deletes the DRM user assertion when the user logs out.

8.3.2 Acquiring User Role Assertions

When a user requests the creation of a new workflow as described in Section 8.1.1 above, the workflow management system must check that the proposed subject of the workflow is a member of the appropriate role. It does this by obtaining a user role assertion from the identity management system as follows:

1. The workflow management system establishes a secure authenticated channel to the identity management system.

2. The workflow management system transmits the identity of the proposed subject to the identity management system.

3. The identity management system creates a user role assertion for that user and transmits it to the workflow management system.

4. The workflow management system checks that the proposed subject is a member of the appropriate role.

8.4 Record Packaging

8.4.1 Retrieving Records

A DRM agent may request a copy of a record at any time by supplying the identifier of that record to the record packager, as follows:

1. The DRM agent transmits the record identifier to the record packager[4].

2. The record packager establishes a secure authenticated channel to the global electronic healthcare infrastructure, and requests a copy of the record.

[4]It may be desirable for the DRM agent to authenticate itself first to prevent spurious requests, but it is not strictly necessary since the record, when returned, will be encrypted.

3. The electronic healthcare record infrastructure transmits a plaintext copy of the record to the record packager.

4. If the record packager has never seen this record before, it chooses a new random record encryption key for it; otherwise it looks up the existing key for this record[5].

5. The record packager encrypts the record using the record encryption key.

6. The record packager transmits the encrypted record to the DRM agent.

8.4.2 Creating and Modifying Records

For simplicity, we will require devices to implement a "write-through" regime for the `create`, `modify` and `annotate` permissions, that is, any modifications or additions to a record must be immediately transmitted back to the global electronic healthcare record infrastructure. Thus the global infrastructure will always contain the most recent version of the record.

Since the global infrastructure may not recognise the device or any licences issued to it, the local facility's record packager will act as an intermediary between its devices and the global infrastructure as follows:

1. The DRM agent establishes a secure authenticated channel to the record packager.

2. The DRM agent transmits the creation or modification licence[6] to the record packager.

3. The DRM agent transmits the new version of the healthcare record and any new or modified attributes to the record packager.

4. The record packager establishes a secure authenticated channel to the global electronic healthcare record infrastructure.

5. The record packager transmits the new version of the healthcare record and its attributes to the global infrastructure.

6. The global infrastructure replaces its version of the record (if any) with the new version, and similarly replaces any old attributes with the values in the request. Attributes that are not present in the request remain unchanged.

Note that we are supposing that the global electronic healthcare record infrastructure trusts local record packagers (and, by extension, local policy officers) to use the modification facility in a responsible way.

[5] or we could adopt SITDRM's method and make the record encryption key into a function of the record identifier and a secret master key.

[6] We will suppose that the record packager trusts the DRM agent without verifying the licence itself, as for workflow initiation earlier.

8.5 Over-ride

We will allow the consent directives stored in the consent directive management system to be over-ridden by making a request to the licence issuer of a facility, which in turn forwards the request to the global consent directive management system. If accepted, the request is served by the global electronic healthcare record infrastructure and facility's record packager as usual.

8.5.1 Over-ride with Consent

A patient (or substitute decision maker) can over-ride his or her own consent directive by requesting a facility's licence issuer to issue a licence for the healthcare worker in question. The facility's licence issuer must first verify that the patient does, in fact, have authority to over-ride the consent directive in question, and then construct and issue an appropriate licence:

1. The patient establishes a secure authenticated channel to the facility's licence issuer[7].
2. The licence issuer obtains the patient's credentials and forwards these to the global consent directive management system over a secure authenticated channel.
3. The consent directive management system verifies the credentials and returns a response to the licence issuer.
4. If successful, the licence issuer obtains the identifier of the DRM agent (or possibly session) used by the healthcare worker to whom the over-ride is being granted, and constructs a time-limited licence for this DRM agent.
5. The licence issuer requests the facility's record packager to obtain and create a protected version of the healthcare record as usual.
6. The licence issuer forwards the licence and the protected record to the DRM agent as usual.

If a patient wishes to permanently alter his or her consent directive, he or she must execute the usual procedure for altering consent directives described in Chapter 3.

8.5.2 Over-ride without Consent

Over-riding a consent directive without the consent of the patient is similar to the procedure with consent, but the healthcare worker him- or herself initiates the request instead of the patient. The consent directive management system may refuse the request if the relevent consent directive specifies that it is never to be over-ridden, or if the consent directive management system deems the request to be unauthorised (for example, the requester does not have this privilege).

[7] Only the licence issuer can be authenticated at this point since it has no credentials for the patient

In a real system, the requester must provide some reason for invoking the over-ride procedure, and all requests and reasons must be logged and made available for auditing purposes. For our purposes, we will suppose that all over-ride requests are logged by the consent directive management system and linked to the consent directive to which they applied. (They might also be logged by the electronic healthcare record infrastructure and linked to the relevent healthcare record, but this requires an extra step to transmit the request from the consent directive management system to the electronic healthcare record infrastructure).

Chapter 9

Security

Our system aims to permit access to personal health information only if the access is required by accepted medical practice according to the principle of least privileges, and

- in a general consent regime, has the consent of the subject of that information (or his or her substitute decision maker); or
- in a general denial regime, has not been prohibited by the subject of that information (or his or her substitute decision maker).

For ease of exposition, we will henceforth use "consent has been granted" and similar expressions to include "consent has not been denied" in a general denial regime, along with its obvious meaning in a general consent regime. We will also assume "patients' consent" to include consent given by their legitimate substitute decision makers.

In this chapter, we describe the conditions that we require in order for the system to be secure, and describe the security properties that we intend each component of the system to have. If all of these requirements are satisfied, we expect that the system will enforce the principle of least privileges and patient consent as desired.

The security of the system depends to a large degree on the trustworthiness and reliability of both its administrators and users. We cannot coerce policy officers into creating good policies, and have only limited ability to coerce good behaviour from users. In particular, our system cannot prevent a number of physical attacks (or accidents) made by insiders, including:

- attacks in which an unauthorised person obtains access to a device that has been authenticated as being operated by an authorised person, such as "shoulder-surfing" attacks or instances in which the authorised person passes a device to an unauthorised person after authenticating him- or herself;
- attacks in which an authorised person records information in an analogue form after being granted access to it, such as writing it down by hand or passing it on by word of mouth; and
- attacks in which a privileged person (such as a facility's policy officer) does not behave in accordance with the requirements we set out below.

9.1 Overview

In our system, the policy officer of a facility acts as the source of accepted medical practice by providing the workflows, authorisation templates, and user-role mappings that describe this practice and the authorisation policy that it implies. The consent directive management system acts as the source of patient consent.

Policies and consent directives are ultimately enforced by the digital rights management system. It is the role of the record packager to translate plaintext health information into a form in which it can only be accessed by the digital rights management system.

It is the role of the licence issuer to translate policies obtained from the workflow management system (and ultimately the policy officer) and consent directive management system into licences that can be directly enforced by the digital rights management system. Thus we require that the licence issuer only issue licences in accordance with the policy provided by these entities.

It is the role of the identity management system, together with the domain and session controller, to translate policies that refer human users and roles into domains and sessions of the digital rights management system. Thus we require that the identity management system only issue DRM user assertions after authenticating both user and DRM agent, and the domain and session controllers to only admit DRM agents to (role) domains if they can provide an appropriate DRM user assertion.

9.2 Consent Directive Management System

The consent directive management system is required to approve the issuing of a licence if and only if either

- consent has been granted for the action described by the licence; or
- the licence is the result of a properly-constituted emergency over-ride request, as described in Section 8.5.

Note that, in the present version of the system, we are not specifically concerned with the confidentiality of consent directives, but only with the integrity of decisions made according to them. We may consider confidentiality in a future version of the system.

We assume that only genuine consent directives created by the relevent patients are inserted into the system; the procedure by which this happens is beyond the scope of the present document. We also assume that the consent directive management system is trusted to apply these directive correctly in response to an access request, as described in Chapter 3.

We assume that the consent directive management system has some method of authenticating all of the licence issuers of all of facilities for which the system is responsible, and that it is consequently able to establish secure authenticated channels with these licence issuers.

We require that consent directive management system only accept a request to issue a licence if it is from one of the licence issuers described above, and to only approve

the request if it is consistent with the aims described above. In this way, the consent directive management system will only give permission to issue a licence if the licence is consistent with all of the legitimate consent directives and any relevent jurisdictional information, according to the algorithm described in Chapter 3, and only authorised licence issuers may obtain this permission.

9.3 Electronic Healthcare Record Infrastructure

The electronic healthcare record infrastructure is required to enforce an access control regime such that

- the record packagers of recognised health facilities have both read and write access to records and resource attributes in the infrastructure; and
- the licence issuers of recognised health facilities have read access to the resource attributes of records in the infrastructure.

No other entities are permitted to have any access to the infrastructure at all.

We assume that the electronic healthcare record infrastructure has some method of authenticating all of the record packagers and licence issuers described above, and that it is consequently able to establish secure authenticated channels with these record packagers and licence issuers.

We require that all communication between the electronic healthcare record infrastructure by conducted over the aforementioned secure authenticated channels. Thus only the record packagers and licence issuers of recognised health facilities may read or write information from or to the database, as described above.

The electronic healthcare record infrastructure does not itself place any limits on which records the record packagers can read or write, or on which attributes can be read by licence issuers. The security and integrity of the system therefore relies on the good behaviour of record packagers and licence issuers, which will be discussed in Section 9.5 below.

9.4 Workflow Management System

The workflow management system is required to

- provide workflows and corresponding authorisation templates in line with accepted medical practice; and
- accurately provide and update the state of a workflow according to requests from DRM agents that are the subjects of that workflow.

(The workflow management system also acts as the session controller in the description of the system given in earlier chapters, but we will consider this role separately in Section 9.5 below.)

We assume that only the policy officer of an organisation is permitted to create and modify workflow descriptions and authorisation templates; the mechanism by which

this is done is beyond the scope of the present document. We assume that this policy officer is trusted to create workflows that achieve their stated purposes in accordance with medical conventions, and to create corresponding authorisation templates in accordance with the principle of least privileges.

We require that the workflow management system only create a new instance of a workflow if it is presented with a valid licence granting the `execute` permission over that workflow, as described in Section 8.1 earlier. We require the workflow management system to check that the proposed subject of the workflow and any resources to be used by the workflow are consistent with both the licence used to initiate the workflow, and the authorisation templates of the workflow.

We also require that the workflow management system be able to authenticate all of the DRM agents within the facility, and that it will only update workflow information in response to properly-constituted requests from DRM agents that are members of the session associated with that workflow.

In this way, workflow instances will only be created if there is a legitimate need for them according to the facility's policy officer, and, by our assumption, only if there is an accepted medical need for them. Furthermore, subjects and resources can only be appointed to workflows in accordance with the policy established by the policy officer. Workflow state information will only be altered according to the requests of DRM agents that have been authenticated as being the subjects of that workflow.

9.5 Digital Rights Management System

The digital rights management system is required to ensure that electronic health records obtained from the electronic healthcare record infrastructure can only be accessed by legitimate DRM agents in accordance with the policies given by the workflow management system of the facility and the global consent directive management system.

The security of the cryptographic protocols, DRM agents, key management and content encryption used in our system is essentially the same as that of OMA DRM (given that sessions are cryptographically identical to domains), and is discussed in detail in the OMA DRM specification [37]. We will therefore assume that

- protected records created by the record packager of a facility will only be used in accordance with licences issued by the licence issuer of the facility; and
- DRM agents may only use licences issued to a domain or session if they have been made a member of that domain or session by the relevent controller.

We additionally assume that devices

- are able to correctly test the `purpose` constraint by comparing the contents of the constraint with the current workflow state as described in Chapter 6;
- leave all domains and sessions, and delete all DRM user assertions, when their human user logs out.

In order to achieve the aims described above, we therefore require that

- the record packager of a facility only distributed records in a protected form;
- the licence issuer of a facility only issue licences in accordance with the workflow management system and consent directive management system; and
- the domain and session controller of a facility only admit DRM agents into a domain or session if they have a legitimate reason to belong to that domain or session.

Record Packager. We require that the record packager of a facility only distribute records in the encrypted form described in Chapter 6, so that records can only be accessed in accordance with a valid licence as discussed there and in the OMA DRM specification. We further require that record packager of a facility only write to records in the global electronic healthcare record infrastructure if it is presented with a valid licence to do so, as described in Section 8.4.2 earlier. In this way, records may only be accessed and modified according to the policy of the licence issuer.

Licence Issuer. We require that the licence issuer only issue licences that are described by the policy officer of a facility (either in the form of authorisation templates of the workflow, or directly), and that have been approved by the global consent directive management system according to the algorithm described in Chapter 3. If this is so, valid licences will only exist if they are consistent with the principle of least privileges as determined by the facility's policy officer, and the relevent consent directives.

Domain and Session Controllers. In our system, the only kind of domains and sessions are role domains and sessions, of which DRM agents may become members only if their current user is a member of the role represented by that domain or session. We therefore require that the domain and session controller admit a DRM agent to a domain or session only if it presents a valid DRM user assertion indicating that its current user is a member of the corresponding role. We assume that the domain and session controller is able to authenticate these assertions.

9.6 Identity Management System

The identity management system is required to

- issue DRM user assertions that identify the current human user of a particular DRM agent, and the roles that that user has activated; and
- issue user role assertions that identify the roles of which a particular human user is a member,

as described in Section 8.3 earlier. We require that identity management system only permit users to activate roles of which they are members.

We assume that only the policy officer a facility is able to create credentials and to assign users to roles; the process by which this is done is beyond the scope of the

present document. We also assume that the policy officer is trusted to assign credentials and roles in a manner consistent with accepted medical practice.

We assume that the identity management system has some method of authenticating all of the DRM agents in the facility. We require that it provide DRM user assertions only after authenticating the requesting DRM agent, and after checking that the credentials that it provides match those of the human user that it claims to be representing. Thus, DRM user assertions may only be obtained by authenticated DRM agents and only for the user who is using that device at the time the request is made.

Chapter 10

Future Work

The system described in this document is intended to provide the basic functionality of a secure electronic healthcare record infrastructure, which enables healthcare workers to carry out their functions while allowing patients to grant or deny consent to access their records in a fairly coarse way. There are a number of more advanced features that might be desirable in a secure electronic healthcare record system, including

- exchange of information and consent directives between separate (e.g. regional or provincial) electronic healthcare record infrastructures;
- partial masking of healthcare records, in which only a portion of a particular healthcare record is the subject of a consent directive;
- an "electronic healthcare portal" that allows patients to view their own healthcare information – and possibly modify their consent directives – through an Internet site or similar;
- the export of anonymised data for research purposes; and
- obligations returned by the consent directive management system.

Bibliography

[1] A. Arnab and A. Hutchison. Persistent access control: A formal model for DRM. In *ACM Workshop on Digital Rights Management*, pages 41–53, Alexandria, Virginia, USA, 2007.

[2] V. Atluri and W.-K. Huang. An authorization model for workflows. In *European Symposium on Research in Computer Security*, pages 44–64, Rome, Italy, 1996.

[3] J. Bergmann, O. J. Bott, D. P. Pretschner, and R. Haux. An e-consent-based shared EHR system architecture for integrated healthcare networks. *International Journal of Medical Informatics*, 76:130–136, 2007.

[4] R. Bhatti, A. Samuel, M. Y. Eltabakh, H. Amjad, and A. Ghafoor. Engineering a policy-based system for federated healthcare databases. *IEEE Transactions on Knowledge and Data Engineering*, 19(9), 2007.

[5] S. Braghin, A. Coen-Porisini, P. Colombo, S. Sicari, and A. Trombetta. Introducing privacy in a hospital information system. In *International Workshop on Software Engineering for Secure Systems*, pages 9–16, Leipzig, Germany, 2008.

[6] M. Casassa Mont and R. Thyne. A systemic approach to automate privacy policy enforcement in enterprises. In *International Workshop on Privacy Enhancing Technologies*, pages 118–134, Cambridge, UK, 2006.

[7] F. Casati, S. Ilnicki, L. Jin, V. Krishnamoorthy, and M.-C. Shan. Adaptive and dynamic service composition in eFlow. In *Conference on Advanced Information Systems Engineering*, pages 13–31, 2000.

[8] S. Chaari, F. Biennier, C. Ben Amar, and J. Favrel. An authorization and access control model for workflow. In *International Symposium on Control, Communications and Signal Processing*, pages 141–148, 2004.

[9] Z. Chen, T. Luo, L. Shi, and F. Hong. Perti (sic) net-based workflow access control model. *Journal of Shanghai University (English Edition)*, 8(1):63–69, 2004.

[10] A. H. Chinaei and F. W. Tompa. User-managed access control for health care systems. In *Secure Data Management*, pages 63–72, 2005.

[11] S.-C. Chou, A.-F. Liu, and C.-J. Wu. Preventing information leakage within workflows that execute among competing organizations. *Journal of Systems and Software*, 75:109–123, 2004.

[12] R. Clarke. e-consent: A critical element of trust in e-business. In *Bled Electronic Commerce Conference*, 2002.

[13] ContentGuard. Extensible Rights Markup Language. `http://www.xrml.org`, 2004.

[14] K. Cook. Evaluating acute abdominal pain in adults. *Journal of the American Academy of Physician Assistants*, 1 March 2005. `http://www.jaapa.com/issues/j20050301/articles/belly0305.htm`.

[15] M. A. C. Dekker and S. Etalle. Audit-based access control for electronic health records. *Electronic Notes in Theoretical Computer Science*, 168:221–236, 2007.

[16] D. Domingos, A. Rito-Silva, and P. Veiga. Authorization and access control in adaptive workflows. In *European Symposium on Research in Computer Security*, pages 23–38, 2003.

[17] T. C. Du, E. Y. Li, and J. W. Wong. Document access control in organizational workflows. *International Journal of Information and Computer Security*, 1(4):437–454, 2007.

[18] A. A. El Kalam and Y. Deswarte. Security model for health care computing and communication systems. In *Security and Privacy in the Age of Uncertainty*, pages 277–288, 2003.

[19] L. Franco, T. Sahama, and P. Croll. Security Enhanced Linux to enforce mandatory access control in health information systems. In *Australasian Workshop on Health Data and Knowledge Management*, pages 27–34, Wollongong, Australia, 2008.

[20] C. K. Georgiadis, I. K. Mavridis, G. Nikolakopoulou, and G. I. Pangalos. Implementing context and team based access control in healthcare intranets. *Medical Informatics*, 27(3):185–201, 2002.

[21] M. Hafner, M. Memon, and M. Alam. Modeling and enforcing access control policies in healthcare systems with SECTET. In *Models in Software Engineering*, pages 132–144, 2008.

[22] P. C. K. Hung and K. Karlapalem. A secure workflow model. In *Australasian Information Security Workshop*, pages 33–41, Adelaide, Australia, 2003.

[23] P. C. K. Hung and Y. Zheng. Privacy access control model for aggregated e-health services. In *International IEEE EDOC Conference Workshop*, pages 12–19, 2007.

[24] M. H. Kang, J. S. Park, and J. N. Froscher. Access control mechanisms for inter-organizational workflow. In *ACM Symposium on Access Control Methods and Technologies*, pages 66–74, Chantilly, Virginia, USA, 2001.

[25] G. Karjoth, M. Schunter, and M. Waidner. Platform for enterprise privacy practices: Privacy-enabled management of customer data. In *International Workshop on Privacy Enhancing Technologies*, pages 69–84, San Francisco, USA, 2002.

[26] S. Kenny and L. Korba. Applying digital rights management systems to privacy rights. *Computers & Security*, 21:648–664, 2002.

[27] K. Knorr. Dynamic access control through Petri net workflows. In *Annual Computer Security Applications Conference*, pages 159–167, 2000.

[28] G. Lee, W. Kim, and D.-K. Kim. A novel method to support user's consent in usage control for stable trust in e-business. In *International Conference on Computational Science and its Applications*, pages 906–914, 2004.

[29] X. Liao, L. Zhang, and S. C. F. Chan. A task-oriented access control model for WfMS. In *Information Security Practice and Experience Conference*, pages 168–177, Singapore, 2005.

[30] D.-R. Liu, M.-Y. Wu, and S.-T. Lee. Role-based authorization for workflow systems in support of task-based separation of duty. *Journal of Systems and Software*, 73:375–387, 2004.

[31] Q. Liu, R. Safavi-Naini, and N. P. Sheppard. Digital rights management for content distribution. In *Australasian Information Security Workshop*, pages 49–58, Adelaide, Australia, 2003.

[32] M. Lyver. Consent directive management framework. Document IG5SA0902-0002, iEHR Technical Project, Canada Health Infoway, 12 August 2007.

[33] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 internet public key infrastructure: Online certificate status protocol – OCSP. Request for Comments 2560, Internet Engineering Task Force, June 1999.

[34] S. Nepal, J. Zic, F. Jaccard, and G. Kraehenbuehl. A tag-based data model for privacy-preserving medical applications. In *International Conference on Extending Database Technology – Workshops*, pages 433–444, Munich, Germany, 2006.

[35] C. M. O'Keefe, P. Greenfield, and A. Goodchild. A decentralised approach to electronic consent and health information access control. *Journal of Research and Practice in Information Technology*, 37(2):161–178, 2005.

[36] Open Digital Rights Language Initiative. The Open Digital Rights Language Initiative. `http://odrl.net`, 2004.

[37] Open Mobile Alliance. DRM specification: Approved version 2.1. `http://www.openmobilealliance.org/Technical/release_program/drm_v2_1.aspx`, 6 November 2008.

[38] Organization for the Advancement of Structured Information Standards. OASIS eXtensible access control markup language (XACML) TC. `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml`, 2008.

[39] Organization for the Advancement of Structured Information Standards. OASIS security services (SAML) TC. `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security`, 2008.

[40] M. Petković, S. Katzenbeisser, and K. Kursawe. Rights management technologies: A good choice for securing electronic health records? In *Securing Electronic Business Processes*, pages 178–187, Warsaw, Poland, 2007.

[41] M. Predeschly, P. Dadam, and H. Acker. Security challenges in adaptive e-health processes. In *International Conference on Computer Safety, Reliability and Security*, pages 181–192, 2008.

[42] J. Qiu and C.-H. Ma. A flexible access control model for workflows. In *International Conference on Computer-Supported Collaborative Work in Design*, pages 606–612, 2008.

[43] J. Reid, I. Cheong, M. Henricksen, and J. Smith. A novel use of RBAC to protect privacy in distributed health care information systems. In *Australasian Conference on Information Security and Privacy*, pages 403–415, 2003.

[44] L. Røstad. An initial model and a discussion of access control in patient controlled health records. In *International Conference on Availability, Reliability and Security*, pages 935–943, 2008.

[45] G. Russello, C. Dong, and N. Dulay. A workflow-based access control framework for e-health applications. In *International Conference on Advanced Information Networking and Applications – Workshops*, pages 111–120, 2008.

[46] H. Schulzrinne and E. Wedlund. Application-layer mobility using SIP. *Mobile Computing and Communications Review*, 4(3):47–57, 2000.

[47] N. P. Sheppard and R. Safavi-Naini. Protecting privacy with the MPEG-21 IPMP framework. In *Workshop on Privacy Enhancing Technologies*, pages 152–171, Cambridge, UK, 2006.

[48] N. P. Sheppard and R. Safavi-Naini. Sharing digital rights with domain licensing. In *ACM Workshop on Multimedia Content Protection and Security*, pages 3–12, Santa Barbara, California, USA, 2006.

[49] H. Song, K. T. Win, and P. Croll. Patient e-consent mechanism: Models and technologies. In *CollECTeR*, 2002.

[50] R. K. Thomas. Team-based access control (TMAC): a primitive for applying role-based access controls in collaborative environments. In *ACM Workshop on Role-Based Access Control*, pages 13–19, 1997.

[51] W. M. P. van der Aalst and A. H. M. ter Hofstede. YAWL: Yet another workflow language. *Information Systems*, 20(4):245–275, 2005.

[52] B. Weber, M. Reichart, W. Wild, and S. Rinderle. Balancing flexibility and security in adaptive process management systems. In *CoopIS/DOA/ODBASE*, pages 59–76, 2005.

[53] K. T. Win and J. A. Fulcher. Consent mechanisms for electronic health record systems: A simple yet unresolved issue. *Journal of Medical Systems*, 31:91–96, 2007.

[54] S. Wu, A. Sheth, J. Miller, and Z. Luo. Authorization and access control of application data in workflow systems. *Journal of Intelligent Information Systems*, 18:71–94, 2002.

[55] Y. Wu, Z. Lv, J. Gu, and W. Zhang. Dynamic and secure control model for workflow management system. In *International Conference on Advanced Language Processing and Web Information Technology*, pages 578–582, 2007.

[56] G.-L. Xing, F. Hong, and H. Cai. A workflow authorization model based on credentials. *Wuhan University Journal of Natural Sciences*, 11(1):198–202, 2006.

[57] L. Yang, Y. Choi, C. Myeonggil, and X. Zhao. FWAM: A flexible workflow authorization model using extended RBAC. In *International Conference on Computer-Supported Collaborative Work in Design*, pages 625–629, 2008.

[58] YAWL. YAWL: Yet another workflow language. `http://www.yawl-system.com`, 2008.

Appendix A

Walk-through

In this appendix, we will describe an example sequence of events in a (very simple) healthcare facility in which the system described in the main body of this document has been installed. We will adopt the example of the diagnosis of acute abdominal pain used by Russello, et al. [45], which is itself based on a procedure described by Cook [14]. The technical details of consent directives, licences, assertions, etc. used in the example will be given in Appendix B.

A.1 Set-up

Before the system can be used, the policy officer of the facility must design all of the workflows required to accomplish all of the tasks carried out within the facility, together with the security policies that go with those workflows. We suppose that the policy offier uses some tool for constructing workflows, and for deriving the authorisation templates that these workflows imply using the principle of least privileges. An authorisation template for the first step of the workflow of Figure A.2, which requires an intern to read a patient's healthcare record, is given in Section B.2.2.

All of the employees of the facility must be enrolled in the facility's identity management system by associating them with an identifier and a set of credentials (e.g. password) by which they can be authenticated. The policy officer must also create all of the roles that will be used in the facility, and assign employees to roles according to the nature of their position in the organisation. We suppose that this is done using some tool of the identity management system.

Our simple facility has only two roles, a *receptionist* and an *intern*. Members of these two roles interact according to three workflows, which will be described individually in the following sub-sections.

A.2 Reception

We will suppose that patients suffering from abodminal pain present themselves to the reception of the facility, where a receptionist assigns them to an intern for diagnosis.

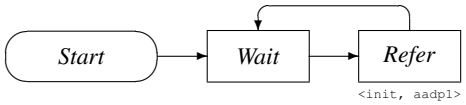

Figure A.1: The workflow for a receptionist. The expression in angle brackets indicates the actions required by the corresponding state of the workflow.

Our simple receptionist has a workflow with only two states, called *wait* and *refer*, that correspond to waiting for patients to arrive and to referring them to an intern, respectively. We suppose that the workflow is initiated by some appropiately authorised actor who does not otherwise appear in this scenario. The workflow is shown diagrammatically in Figure A.1.

While he or she is waiting for patients, the receptionist does not have any permission to do anything. When a patient arrives, however, the receptionist indicates the arrival to her computer. The computer then requests the workflow management system to advance to the *refer* state.

In the *refer* state, the receptionist gains the permission to initiate a diagnosis workflow (called `aadp1`). The technical details of the licence that permit this are given in Section B.3.1. The receptionist chooses an intern who is available to do the diagnosis, and requests the workflow management system to create a new workflow with this intern as the subject, and the new patient's healthcare record as the resource.

Before the workflow can begin, however, the facility's licence issuer must check that consent exists for all of the actions required by the workflow. It does this by obtaining all of the proposed actions from the facility's workflow management system, then sending them as a request to the global consent directive management system as described in Section 8.1.3. The consent directive management system compares the proposed actions with the relevent consent directives and returns a response. If the response is positive, the workflow can be started and executes as described in Section A.3 below.

If the response from the global consent directive management system is negative, however, the licence issuer must refuse permission for the workflow to begin. Depending on the reason for the refusal, the receptionist may have to

- assign the patient to a different intern (if the chosen intern is barred from that patient's health record for some reason);
- ask the patient to suspend his or her consent directive using the over-ride procedure described in Section 8.5.1;
- ask someone with authority to over-ride the patient's consent directive using the procedure described in Section 8.5.2 if this seems justified; or
- turn the patient away (which is hopefully unlikely).

Once the patient has been assigned to an intern for diagnosis, the receptionist returns to the *wait* state until the next patient arrives.

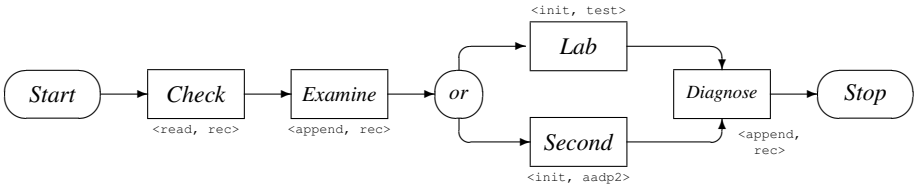

Figure A.2: The workflow for obtaining a diagnosis, adapted from [45].

A.3 Diagnosis

The workflow for an intern charged with diagnosing abdominal pain is shown diagrammatically in Figure A.2. It has five states, called *check*, *examine*, *lab*, *second* and *diagnose*.

The intern's first step, represented by the *check* state of the workflow, is to check the patient's existing healthcare record. In this state, therefore, the intern will be granted permission to read (only) the relevent healthcare record. A licence that allows this is described in detail in Section B.3.2.

Having checked the existing healthcare record, the intern indicates to the workflow management system that he or she is ready to conduct a physical examination. The workflow advances to the *examine* state, and the intern will gain permission to append information to the healthcare record of the patient. The intern can then make the examination and append the results to the record. The modified record will be immediately transmitted back to the global electronic healthcare record infrastructure.

The intern must now choose whether to ask for a lab test, or to seek a second opinion from another intern. The workflow will advance to either the *lab* or *second* states accordingly.

In the *lab* state, the intern gains permission to invoke a service of the lab, called `test`. We suppose that the lab involved is an external organisation, and that invoking the `test` service causes a new worklow instance to be created in that organisation. We will discuss this workflow further in Section A.4 below.

In the *second* state, the intern gains permission to initiate a workflow within his or her own organisation, called `aadp2`. This works much the same way as for the receptionist in Section A.2 above: the first intern selects a second intern who is available to give a second opinion, and requests a new `aadp2` workflow instance to be created with that intern as its subject and the patient's healthcare record as its resource. We will discuss this workflow further in Section A.5 below.

In either case, the first intern must wait until the `test` or `aadp2` workflow completes before his or her own workflow will advance to the *diagnose* state. In this state, the intern re-gains permission to append information to the patient's healthcare record, and he or she will append a final diagnosis to the record. Again, the updated record is immediately stored in the global electronic healthcare record infrastructure.

In a real facility, the patient would presumably be sent somewhere for treatment at this point, but we will stop here.

A.4 Lab Test

If the original intern chooses to seek a lab test, he or she will present his or her credentials to the lab and request the necessary service. Supposing that the credentials are accepted, the lab's workflow management system will create a new instance of the corresponding workflow and assign a subject to it (presumably some employee of the lab).

Before the new workflow can be started, the lab's licence issuer must check that the patient has given consent for all of the actions entailed by that workflow. This is done in the same way as it is done for the workflow initiated by the original receptionist in Section A.2 above.

A.5 Second Opinion

If the original intern chooses to seek a second opinion, he or she will request his or her own facility's workflow management system to create a new instance of the `aadp2` workflow for obtaining a second opinion. We suppose that this workflow is the same as the one for obtaining a first opinion shown in Figure A.2, except that there is no option to seek a lab test or second (actually third) opinion, that is, the *examine* state is followed directly by the *diagnose* state.

As for the other workflows, the licence issuer of the facility must check that consent exists for all of the actions required by the workflow before it can be initiated. If this is successful, the second intern may carry out the workflow in the same manner as the first intern.

Appendix B

Examples

B.1 Consent Directives

B.1.1 Consent Directive

The following simple consent directive grants consent to employees of Facility A to access any healthcare records that refer to Bob (identified as `urn:ca:health:patients:bob`), for the purpose of diagnosing an illness. This purpose is understood by the global consent directive management system by an identifier `urn:ca:health:purposes:diagnose`. The values of the `MatchId` and `DataType` attributes have been abbreviated for clarity.

```
<Rule RuleId="rule1" Effect="Permit">
  <Target>

    <Subjects>
      <Subject>
        <SubjectMatch MatchId="string-equal">
          <AttributeValue DataType="string">
            urn:ca:health:facilities:A
          </AttributeValue>
          <SubjectAttributeDesignator AttributeId="facility-id"
              DataType="string"/>
        </SubjectMatch>
      </Subject>
    </Subjects>

    <Resources>
      <Resource>
        <ResourceMatch MatchId="string-equal">
          <AttributeValue DataType="string">
            urn:ca:health:patients:bob
```

```
          </AttributeValue>
          <ResourceAttributeDesignator AttributeId="subject-id">
              DataType="string"/>
        </ResourceMatch>
      </Resource>
    </Resources>

    <Actions>
      <Action>
        <ActionMatch MatchId="string-equal">
          <AttributeValue DataType="string">
            urn:ca:health:purposes:diagnose
          </AttributeValue>
          <ActionAttributeDesignator AttributeId="purpose"
              DataType="string"/>
        </ActionMatch>
      </Action>
    </Actions>

  </Target>
</Rule>
```

B.1.2 Consent Request

The following request is for access to a healthcare record with identifier `urn:ca:health:ehr:bob` (Bob's healthcare record). The person requesting access is Alice, identified as `urn:ca:health:people:alice`, who is an intern at Facility A. Alice is seeking access to the healthcare record for the purpose of diagnosing an illness. This request might result from the desire to issue the licence shown in Section B.3.2, for example. Note that the licence issuer must obtain the value of the `subject-id` resource attribute from the global electronic healthcare record infrastructure prior to forming the request, since this attribute is not available from the authorisation template.

```
<Request>

  <Subject>
    <Attribute AttributeId="subject-id">
      <AttributeValue>
        urn:ca:health:people:alice
      </AttributeValue>
    </Attribute>
    <Attribute AttributeId="facility-id">
      <AttributeValue>
        urn:ca:health:facilities:A
      </AttributeValue>
```

```
    </Attribute>
    <Attribute AttributeId="role-id">
      <AttributeValue>
        urn:ca:health:roles:intern
      </AttributeValue>
    </Attribute>
  </Subject>

  <Resource>
    <Attribute AttributeId="resource-id">
      <AttributeValue>
        urn:ca:health:ehr:bob
      </AttributeValue>
    </Attribute>
    <Attribute AttributeId="subject-id">
      <AttributeValue>
        urn:ca:health:patients:bob
      </AttributeValue>
    </Attribute>
  </Resource>

  <Action>
    <Attribute AttributeValue="purpose">
      <AttributeValue>
        urn:ca:health:purposes:diagnose
      </AttributeValue>
    </Attribute>
  </Action>

</Request>
```

B.2 Workflows

B.2.1 Workflow Description

Unlike the other languages used in this appendix, YAWL workflows are visually represented as graphs rather than text files. Appendix A gives several examples.

B.2.2 Authorisation Template

The following authorisation template is for the *check* step of the workflow for diagnosing acute abdominal pain used in Appendix A. This step requires the subject of the workflow (who must be an intern) to read an electronic healthcare record. The record can be referred to in other documents by the variable name `ehr`.

```
<o-t:template>
  <o-ex:agreement>

    <o-ex:party>
      <o-t:forany>
        <o-ex:context>
          <o-dd:role>
            urn:ca:health:roles:intern
          </o-dd:role>
        </o-ex:context>
      </o-t:forany>
    </o-ex:party>

    <o-ex:asset>
      <o-t:forany var="ehr">
        <o-ex:context>
          <o-dd:role>
            urn:ca:health:types:ehr
          </o-dd:role>
        </o-ex:context>
      </o-t:forany>
    </o-ex:asset>

    <o-ex:permission>
      <o-dd:play/>
      <o-ex:constraint>
        <o-dd:purpose>
          urn:wf:aadp1:check
        </o-dd:purpose>
      </o-ex:constraint>
    </o-ex:permission>

  </o-ex:agreement>
</o-t:template>
```

B.3 Digital Rights Management

B.3.1 Workflow Initiation Licence

The following licence permits a member of Carol's personal domain to initiate a workflow identified as `urn:wf:aadp1` (the main workflow for diagnosing acute abdominal pain in Appendix A). The workflow takes one parameter, called `ehr`, for which Carol may use any health record (in practice, she should use the healthcare record that belongs to the patient for which the workflow was created, but there is no way of know-

ing who this might be at the time the licence must be issued). The cryptographic details of the signature have been omitted for brevity.

```
<o-ex:rights>
  <o-ex:agreement>

    <o-ex:party>
      <o-ex:context>
        <o-dd:uid>urn:dom:personal:carol</o-dd:uid>
      </o-ex:context>
    </o-ex:party>

    <o-ex:asset>
      <o-ex:context>
        <o-dd:uid>urn:wf:aadp1</o-dd:uid>
      </o-ex:context>
    </o-ex:asset>

    <o-ex:permission>
      <o-dd:execute/>

      <o-ex:asset param="ehr">
        <o-ex:context>
          <o-dd:role>urn:ca:health:type:ehr</o-dd:role>
        </o-ex:context>
      </o-ex:asset>

    </o-ex:permission>

    <ds:Signature>
      ...
    </ds:Signature>

  </o-ex:agreement>
</o-ex:right>
```

B.3.2 Healthcare Record Licence

Suppose that the `urn:wf:aadp1` workflow is initiated for a patient Bob, according to the initiation licence of Section B.3.1. The workflow management system creates a session – identified as `urn:sess:47` here – and instantiates the authorisation template of Section B.2.2 for this session and Bob's healthcare record. The resulting licence allows members of the new session to `play` (view) Bob's health record, as long as the session is in the `urn:wf:aadp1:check` state of the workflow. The cryptographic details of the record encryption key and the signature have been omitted for brevity.

```
<o-ex:rights>
```

```
  <o-ex:agreement>

    <o-ex:party>
      <o-ex:context>
        <o-dd:role>urn:sess:47</o-dd:role>
      <o-ex:context>
    </o-ex:party>

    <o-ex:asset>
      <o-ex:context>
        <o-dd:uid>urn:ca:health:ehr:bob</o-dd:uid>
      </o-ex:context>
      <ds:KeyInfo>...</ds:KeyInfo>
    </o-ex:asset>

    <o-ex:permission>
      <o-dd:play/>
      <o-ex:constraint>
        <o-dd:purpose>
          urn:wf:aadp1:check
        </o-dd:purpose>
      </o-ex:constraint>
    </o-ex:permission>

    <ds:Signature>
      ...
    </ds:Signature>

  </o-ex:agreement>
</o-ex:rights>
```

B.3.3 Creation Licence

The following licence permits a member of the session identified as `urn:sess:23` to create a new record with unique identifier `urn:ca:health:ehr:bob:xray:34`. The new record is of type `urn:ca:health:types:xray` (that is, an X-ray), and its subject is Bob. Note that the `subject` element is not part of the ODRL standard, but we have used it here to refer to the `subject-id` attribute of a healthcare record.

```
<o-ex:rights>
  <o-ex:agreement>

    <o-ex:party>
      <o-ex:context>
        <o-dd:role>urn:sess:23</o-dd:role>
      <o-ex:context>
    </o-ex:party>
```

```
    <o-ex:asset>
      <o-ex:context>
        <o-dd:uid>urn:ca:health:ehr:bob:xray:34</o-dd:uid>
        <o-dd:role>urn:ca:health:types:xray</o-dd:role>
        <o-dd:subject>urn:ca:health:patients:bob</o-dd:subject>
      </o-ex:context>
    </o-ex:asset>

    <o-ex:permission>
      <o-dd:create/>
    </o-ex:permission>

    <ds:Signature>
      ...
    </ds:Signature>

  </o-ex:agreement>
</o-ex:rights>
```

B.4 Identity Management

B.4.1 DRM User Assertion

The following DRM user assertion asserts that some DRM agent is being used by Alice, and that Alice has activated the "intern" role. The DRM agent is identified by the hash value of its public key, using the format specified by the OMA DRM Rights Object Acquisition Protocol Suite. The DRM agent's user (Alice) and her roles are specified as the values of the `urn:drm:user:user-id` and `urn:ca:health:subject:role-id` attributes, respectively. The assertion was issued by the identity management system of Facility A. The cryptographic details of the signature have been omitted for brevity.

```
<saml:Assertion>

  <saml:Issuer>
    urn:ca:health:facilities:A:idms
  </saml:Issuer>

  <ds:Signature>
    ...
  </ds:Signature>

  <saml:Subject>
    <roap:deviceID>
      <roap:keyIdentifier xsi:type="roap:X509SPKIHash">
        <roap:hash>aXENc+Um/9/NvmYKIhDLaErK0fk=</roap:hash>
```

```
      </roap:keyIdentifier>
    </roap:deviceID>
  </saml:Subject>

  <saml:AttributeStatement>
    <saml:Attribute Name="urn:drm:user:user-id">
      <saml:AttributeValue>
        urn:ca:health:people:alice
      </saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>

  <saml:AttributeStatement>
    <saml:Attribute Name="urn:ca:health:subject:role-id">
      <saml:AttributeValue>
        urn:ca:health:roles:intern
      </saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>

</saml:Assertion>
```

B.4.2 User Role Assertion

The following user role assertion, also made by the identity management of Facility A, asserts that Alice has a subject identifier of `urn:ca:health:people:alice`, is a member of the "intern" role, and works for Facility A.

```
<saml:Assertion>

  <saml:Issuer>
    urn:ca:health:facilities:A:idms
  </saml:Issuer>

  <ds:Signature>
    ...
  </ds:Signature>

  <saml:Subject>
    <saml:NameID>
      urn:ca:health:people:Alice
    </saml:NameID>
  </saml:Subject>

  <saml:AttributeStatement>
```

```
    <saml:Attribute Name="urn:ca:health:subject:subject-id">
      <saml:AttributeValue>
        urn:ca:health:people:alice
      </saml:AttributeValue>
    <saml:Attribute Name="urn:ca:health:subject:role-id">
      <saml:AttributeValue>
        urn:ca:health:roles:intern
      </saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute Name="urn:ca:health:subject:facility-id">
      <saml:AttributeValue>
        urn:ca:health:facilities:A
      </saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>

</saml:Assertion>
```