SECURITY, PRIVACY, IDENTITY AND PATIENT CONSENT MANAGEMENT ACROSS HEALTHCARE ENTERPRISES IN INTEGRATED HEALTHCARE ENTERPRISES (IHE) CROSS ENTERPRISE DOCUMENT SHARING (XDS) AFFINITY DOMAIN

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TUNCAY NAMLI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JUNE 2007

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Volkan Atalay
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Asuman Doğaç
Supervisor

Examining Committee Members

Prof. Dr. Özgür Ulusoy (Bilkent,CENG) _____

Prof. Dr. Asuman Doğaç (METU,CENG) _____

Prof. Dr. İ. Hakkı Toroslu (METU,CENG) _____

Assoc. Prof. Dr. Nihan K. Çiçekli (METU,CENG) _____

Gökçe Banu Laleci Ertürkmen (METU,SRDC) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: TUNCAY NAMLI

Signature:

ABSTRACT

SECURITY, PRIVACY, IDENTITY AND PATIENT CONSENT MANAGEMENT ACROSS HEALTHCARE ENTERPRISES IN INTEGRATED HEALTHCARE ENTERPRISES (IHE) CROSS ENTERPRISE DOCUMENT SHARING (XDS) AFFINITY DOMAIN

Namli, Tuncay

M.Sc., Department of Computer Engineering

Supervisor: Prof. Dr. Asuman Doğaç

June 2007, 69 pages

Integrated Healthcare Enterprise (IHE) is an initiative by industry and healthcare professionals to improve knowledge sharing and interoperability between healthcare related enterprises. IHE publishes Integration Profiles on several Healthcare Fields to define how systems can use existing standards and technologies to execute a specific use case in healthcare. Cross Enterprise Document Sharing (XDS) is such a profile which defines the way of sharing Electronic Health Records (EHR) between healthcare enterprises. In this thesis, IHE Cross Enterprise User Authentication, IHE Node Authentication and Audit Trail, IHE Basic Patient Privacy Consent profiles are implemented based on the IHE XD-Simplementation by National Institute of Standards, USA. Furthermore, some of the unspecified issues related with these profiles are clarified and new techniques are offered for their implementations. One of the contribution of the thesis is to use OASIS Extensible Access Control Markup Language (XACML)

to define patient consent policies and manage access control. Other technologies and standards that are used in the implementation are as follows; OASIS Security Assertion Markup Language (SAML), XML Signature, Mutual Transport Layer Security (TLS), RFC 3195 Reliable Delivery for Syslog, RFC 3881 Security Audit and Access Accountability Message XML Data Definitions.

Keywords: IHE Profiles, IHE XDS, Cross User Authentication, RBAC, Auditing, SAML, XACML

ÖZ

IHE XDS PLATFORMUNDA SAĞLIK SİSTEMLERİ ARASINDA GÜVENLİK, GİZLİLİK, KULLANICI KİMLİĞİ VE HASTA HAKLARI YÖNETİMİ

Namli, Tuncay

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Asuman Doğaç

Haziran 2007, 69 sayfa

Integrated Healthcare Enterprise (IHE) bilgisayar sistemleri üzerinde sağlık alanında bilgi paylaşımını ve birlikte çalışabilirliği sağlamak amacıyla, sağlık uzmanları ve sağlık alanında çalışan endüstri kuruluşları tarafından olusturulan bir teşebbüstür. IHE, "Integration Profile" adı verilen, günümüz standart ve teknolojilerinin sağlık alanındaki belirli senaryoları gerçekleştirmek için nasıl kullanılması gerektigini belirleyen profiller çıkarır. Bunlardan bir tanesi de sağlık hasta kayıtlarının farklı sağlık kuruluşları arasında paylaşılabilmesini sağlayan Cross Enterprise Document Sharing (XDS) profilidir. Bu tezde IHE XDS platformu üzerinde güvenlik, gizlilik ve hasta hakları konularını inceleyen "IHE Cross Enterprise User Authentication", "IHE Node Authentication and Audit Trail", "IHE Basic Patient Privacy Consent" profilleri uygunlanmıştır. Bunlarla beraber hasta gizlilik poliçeleri oluşturmak ve rol bazlı yetkilendirme sisteminde kullanmak için OASIS Extensible Access Control Markup Language (XACML) standardı kullanılmıstır. Ayrıca yazılımda "OASIS Security Assertion Markup

Language (SAML)", "XML Signature (w3c)", "Mutual Transport Layer Security (TLS)", "RFC 3195 Reliable Delivery for Syslog", "RFC 3881 Security Audit and Access Accountability Message XML Data Definitions" gibi standartlar uygulanmıştır.

Anahtar Kelimeler: IHE Profilleri, IHE XDS, Kullanıcı Yetkilendirme, Rol tabanlı ulaşım kontrolü, Güvenlik Denetimi, SAML, XACML

To my family

ACKNOWLEDGMENTS

I am honored to present my special thanks and deepest gratitude to my supervisor Prof. Dr. Asuman DOĞAÇ for all her guidance and support during this work.

I would like to thank Software Research and Development Center team for their support and patience during this study.

Finally, I would like to thank my family for all their life-long support.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The quality of healthcare depends on the existence of accurate health information. The accessibility of healthcare data through networked eHealth applications will change the way healthcare is delivered. Sharing Electronic Healthcare Records (EHRs) and allowing patients to access their medical information and hence having their informed and responsible participation in care processes will bring radical improvements to the quality and efficiency to the European healthcare systems. The fast and ubiquitous access to patient records and other medical information provided by such networks could reduce the number of medical errors due to inadequate information regarding a patient's history, prescribed medication and current condition. Implementing an eHealth network could enable the sharing of medical record information with enough speed and accuracy to be of value to a physician examining an emergency patient at a remote site.

Europe is facing the challenge of delivering quality healthcare to all its citizens, at affordable cost. The accessibility of healthcare data through networked eHealth applications will bring radical improvements to the quality and efficiency to the European healthcare systems. However, the unusually sensitive nature of health information requires that a particular attention must be paid to privacy and security of healthcare data. Without proper privacy and security mechanisms, patients will be reluctant to participate in the electronically

connected health networks.

Currently new networked eHealth applications are emerging: sharing the Electronic Healthcare Records (EHRs) of patients and empowering the patients with access to their Personal Healthcare Records have become global priorities in the healthcare IT domain since effective use of EHRs has the potential to positively influence both the quality and the cost of health care. There are several global initiatives, for example the Healthcare Information Technology Standards Panel (HITSP) implementation specification for EHR interoperability in the USA [1] and Electronic Health Record Solution (EHRS) Blueprint [2] of the Health Infoway in Canada all describe how to network eHealth applications; share EHRs and empower patients. These emerging networked eHealth applications bring about new risks and hence new needs of privacy: These efforts can only be successful if they are complemented with proper privacy and security enhancing tools since patient healthcare records contain extremely sensitive information. In order to protect the privacy of healthcare records and patient privacy following seven privacy principles are determined:

- Openness and Transparency: Individuals should be able to understand what information exists about them, how that information is used, and how they can exercise reasonable control over that information. This transparency helps promote privacy practices and gives confidence to individuals with regard to data privacy, which in turn can help increase participation in digital networks.

- Purpose Specification and Minimization: Data use must be limited to the amount necessary to accomplish specified purposes. Minimization of use will help reduce privacy violations, which can easily occur when data is collected for one legitimate reason and then reused for different or unauthorized purposes.

- Collection Limitation: Personal data should be obtained only by fair and lawful means, and, if applicable, with the knowledge or consent of the perti-

nent individual. In an electronic networked environment, it is particularly important for individuals to understand how information concerning them is being collected because electronic collection methods may be confusing to average users.

- Use Limitation: The use and disclosure of private information should be limited to those purposes specified by the data recipient.

- Individual Participation and Control: Every individual should retain the right to request and receive in a timely and intelligible manner information regarding who has that individual's health data and what specific data the party has, to know any reason for a denial of such request, and to challenge or amend any personal information. Individual participation promotes data quality, privacy, and confidence in privacy practices.

- Data Integrity and Quality: Data should be accurate, complete, relevant, and up-to-date especially in critical domains like eHealth to ensure its usefulness. For instance, the quality of health care depends on the existence of accurate health information.

- Accountability and Oversight: Privacy protections have little weight if privacy violators are not held accountable for compliance failures. Privacy audits and other oversight tools can help to identify and address privacy violations and security breaches by holding accountable those who violate privacy requirements and identifying and correcting weaknesses in their security systems.

Today, most clinical and EHR systems deployed incorporate relatively simple access control measures, usually to support needs within a single organisation. Only a few of these are interoperable across vendor products or with other relevant systems and hence fail to satisfy the new needs.

Identity management, trust and privacy are already active research and development areas especially in the eBusiness domain with solutions addressing

problems like "how to protect the privacy of the user during online operations on the Internet" or "providing optimal match between market offerings and expectations of customers while respecting customer's privacy". However, in the healthcare domain the problem is reversed: the primary concern is not the privacy of the customer, i.e., the physician who is trying to access the patient clinical information but the privacy of the record that he is trying to access.

Integrated Healthcare Enterprise (IHE)[3] is an initiative by industry and healthcare professionals to improve knowledge sharing and interoperability between healthcare related enterprises. IHE publishes Integration Profiles on several Healthcare Fields to define how systems can use existing standards and technologies to execute a specific use case in healthcare. Cross Enterprise Document Sharing (XDS)[4] is such a profile which defines the way of sharing Electronic Health Records (EHR) between healthcare enterprises. IHE XDS becomes very popular in healthcare IT sector and many national healthcare initiatives are using the profile as the basis of their national healthcare network like NHIN in US, Health Infoway in Canada, and Health@net [5] in Austria.

This thesis concentrates on security, privacy and patient consent management while healthcare enterprises share their medical records through an IHE XDS based platform. In this respect, three main issues are addressed in this work; identity management and cross user authentication, role base access control with considering patient consents, and audit trailing.

This thesis is organized as follows: Chapter 2 summarizes the related work by emphasizing the innovative aspects of the proposed solution. In Chapter 3 the main technologies that have been used in this thesis are presented. Chapter 4 is devoted to the description of the overall privacy system developed in the scope of thesis work. Finally Chapter 5 concludes the thesis.

CHAPTER 2

RELATED WORK

In this chapter, the previous work on identity management, privacy and access control, and auditing mechanisms related with the thesis are briefly described. Each section provides information about acadamic, standard, or industry initiatives.

2.1 Identity Management

In 1999 Microsoft introduced Microsoft Passport system which provides single sign-on for web sites. Then, in 2001, Liberty Alliance Project [6] is initiated which broadened the focus of identity management with attribute federation and identity provisioning among more than one service providers. Microsoft has also initiated another project called "TrustBridge" in 2002 in order to provide federation in identity management however not much development has been achieved until now.

Another major initiative on Federated Identity Management is from OASIS Security Services Committee. It has published Security Assertion Markup Language (SAML) [7] V2.0 with the contributions of Liberty Alliance and Shibboleth initiatives. SAML provides XML based framework which allows business entities to make assertions regarding the identity, attributes, and entitlements

of a subject to other entities, such as a partner company or another enterprise application. In this respect, it becomes a base framework for other identity management initiatives like Liberty Alliance and Shibboleth.

The Liberty Alliance is a consortium of more than 150 members consisting of government agencies, enterprise end-users, technology vendors and other type of companies and organizations. The Liberty Alliance consortium is founded in early 2002. It has a vision of a networked world in which the individuals, businesses, organizations and institutions can more easily interact and collaborate with one another while respecting the privacy and security of shared identity information. In this respect, they develop open standards and provide best practice guidance for public policy compliance, privacy concerns, business requirements and interoperability conformance testing and certification.

The Liberty Alliance has divided the process of developing Interoperable Federated Identity Services into three phases and published a set of specifications for each phase:

- *Simplified Single Sign-On and Identity Federation (ID-FF):* The ID-FF specifications provide details about how to achieve identity federation by using some features such as identity linkage, simplified sign-on and simple session management.

- *Web Service Framework (ID-WSF):* The ID-WSF specifications provide a framework for building interoperable identity services, permission based attribute sharing, identity service description and discovery and the associated security profiles.

- *Enhancements to Federation and Services Infrastructure (ID-SIS):* The ID-SIS specifications provide interoperability profiles for special identity services such as personal identity profile service, alert service, calendar service, wallet service, contacts service, geo-location service, presence service and so on.

The Liberty Alliance has formed several special interest groups working on specialized issues and domains. One of them is for Healthcare domain and has the objective of providing a forum for interested parties to discuss the identity management, security and privacy issues specific to healthcare systems.

2.1.1 Identity Management in Web Services

In paralel to those efforts in Federated Identity Management, there is also WS-* family standards which provides identity management solutions for web service environments. The WS-Security specification [8] is the basic building block for the web service security which can be used within several security models (PKI, Kerberos, etc) and supports for multiple security token formats, multiple encryption and signature formats and multiple trust domains. The WS-Security specification provides three main mechanisms:

- Message integrity
- Message confidentiality
- Ability to send security tokens as a part of the message (in SOAP headers)

The WSS specification just specifies how the security tokens are carried with the SOAP request, does not define how the web service provider can establish trust for these security tokens and ensure the identity of the web service requestor. There is other web service standard published for these purpose, the WS-Trust specification [9], which provides a trust model for the web service environments and methods for issuing, renewing and validating security tokens. These security tokens are used for authentication, identity federation, attribute federation and non-repudiation for web service transactions.

Sometimes the Requestor may need a series of transactions (conversation) with Web service. In these situations, the authentication and other identity information should not be provided for each transaction. The WS-SecureConversation specification [10] defines how the web services can establish a security context for

a conversation and how this conversation can be secured. The WS-Federation [11] specification describes how WS-Security and WS-Trust can be combined in order to construct more complex trust models. It gives several scenarios between different trust domains.

2.2 Privacy and Access Control

Access Control is a hot research area for many years. Role based access control (RBAC) [12], as formalized in 1992 by David Ferraiolo and Rick Kuhn, has become the predominant model for advanced access control because it reduces the complexity and cost of security administration in large networked applications. Then on February 11, 2004, the National Institute of Standards and Technology model for RBAC was adopted as an American National Standard by the American National Standards Institute, International Committee for Information Technology Standards (ANSI/INCITS).

OASIS Extensible Access Control Markup Language (XACML) [13] standard and IBM Enterprise Authorization Language (EPAL) [14] are the two major industry specifications for authorization and access control. Both EPAL and XACML share an abstract model for policy enforcement defined by the Internet Engineering Task Force (IETF) [15] and ISO. XACML provides more features like combining result of multiple policies, ability to reference other policies, ability to return separate results for each node when access to a hierarchical resource (fine-grained access control), and support for attribute values that are instances of XML schema elements which are needed for constructing complex policies.

There are also research and development work which offer new access control models: A Generalized Temporal Role-Based Access Control Model (GTRBAC) [16], Access Control in an Open Distributed Environment (OASIS) [17], and Organization Based Access Control (OrBAC) [18]. The GTRBAC model allows expressing periodic as well as duration constraints on roles, user-role assignments, and role-permission assignments. The OASIS model offers a role based access control model which defines how to make role activations with user credentials.

The OrBAC model proposes solutions for contextual permissions, obligations and rules specific to organizations.

Regarding the access control in healthcare and patient consents there are some activities from standard bodies. CEN/TC 251 prEN 13606-4 [19] specifies a consent document which is defined in the EHR Format described in CEN/TC 251 prEN 13606-1 and communicated together with the EHR itself. The pre-standard also defines the sensitivity classifications of EHR data and the functional roles in accessing the EHR data.

IHE has a profile called "Basic Patient Privacy Consents (BPPC)" [20] for addressing privacy concerns of patients when sharing EHRs through XDS. BPPC profile provides a mechanism through which an affinity domain (a set of healthcare institutes that has agreed to share EHRs) can create a basic vocabulary of codes that identify affinity domain privacy consent policies with respect to information sharing. Each privacy consent policy identifies in legal text what are the acceptable re-disclosure uses, which functional roles may access a document and under which conditions. The patients may use wet signatures (ink on paper signature) or if a patient has a Public Certificate, he may use a digital signature to sign policy documents. The profile gives some example EHR sensitivity levels and some functional roles. This profile considers consents in two categories: Implied Consent and Explicit Consent. Implied consent means that the clinical documents would be marked with the general use consent. In the explicit consent case, an actual instance of a Patient Privacy Consent document will be available. IHE BPPC left many issues open:

- How to publish privacy Consent Policies in the affinity domain?

- The mechanism by which consumers associate individual users with functional roles.

- More importantly, there is no mechanism provided for making consents machine processable.

Following standards are published by American Society for Testing and Materials (ASTM) regarding the healthcare privacy.

- E 1762-95 (2003): Standard Guide for Electronic Authentication of Health Care Information
- E 1986-98: Standard Guide for Information Access Privileges to Health Information
- E2084-00: Standard Specification for Authentication of Healthcare Information Using Digital Signatures
- E2085-00a: Standard Guide on Security Framework for Healthcare Information

2.3 Audit Trailing Systems

Audit trail systems are a core topic for information security and accordingly there is a rich body of literature. However, these studies are mostly in operating system level or implemented systems are for single enterprises. Distributed audit service (XDAS) [21] is one of the distributed audit initiative launched by the Open Group consortium. The objective of the XDAS specification is to define a set of generic events of relevance at a global distributed system level, a common portable audit record format to facilitate the merging and analysis of audit information from multiple components at the distributed system level. XDAS is not attempting to domain specific events, and only considers those events of significance at a distributed system level.

In healthcare, how audit logs are implemented is quite specific for each EHR system, partly determined by the persistence (e.g. database storage) approach adopted, and might also partly be directed by local or national legislation. The only standardization activity in this respect in healthcare is Audit Trail and Node Authentication (ATNA) [22] profile of IHE. ATNA profile recommends the usage of "Security Audit and Access Accountability Message XML Data

Definition for Healthcare Applications" specification (IETF RFC-3881) [23] as audit record format. It also recommends "DICOM Supplement 95" [24] which provides event vocabulary for healthcare.

In the communication protocol level, two specification exists: BSD Syslog Protocol (RFC3164) [25] and Reliable Delivery for Syslog (RFC3195) [26]. There are several known limitations of BSD Syslog. For example, there is no confirmation to the sender that the audit record message was received at the destination. Also there are no options to encrypt the audit record messages or authentication by means of certificates of the sending nodes and the central audit repository. Reliable Delivery for Syslog together with these functionalities provides reliability for the transport.

CHAPTER 3

TECHNOLOGIES AND STANDARDS USED

This section provides the technologies, standards and profiles used or implemented in the thesis work. In the first part Integrated Healthcare Enterprise (IHE) and the IHE XDS Integration Profile, to which this thesis is based. In addition, IHE Audit Trail and Node Authentication profile which is implemented in the scope of the thesis to provide an auditing architecture, is briefly described. In the second part, OASIS Security Assertion Markup Language is detailed to prepare the ground for federated identity management. Third part briefly describes OASIS Extensible Access Control Markup Language (XACML) which is used in the thesis work to represent patient consent policies.

3.1 Integrating the Healthcare Enterprise

Integrating the Healthcare Enterprise (IHE) is a non-profit initiative that was founded in 1998 in the USA by the Radiological Society of North America (RSNA) and the Healthcare Information and Management Systems Society (HIMSS). IHE provides the specifications in order to facilitate the integration of healthcare information resources. In fact, IHE does not develop standards, but recommends the appropriate standards for specific cases. IHE initiative uses the existing standards such as HL7, ASTM, DICOM, ISO, IETF, or OASIS rather

than defining new ones. The IHE specifications constrain these standards where necessary for the integration of the different domain elements such as laboratories, departments, or patient identifiers in the healthcare enterprises.

IHE provides a range of different documents called frameworks. The most common ones are IT Infrastructure Technical Framework(ITI-TF), Cardiology Technical Framework, Laboratory Technical Framework, Radiology Technical Framework and Patient Care Coordination Technical Framework.

3.1.1 IHE Cross Enterprise Document Sharing Profile

Cross-Enterprise Document Sharing enables different healthcare organizations in different clinical affinity domains to exchange documents in the care of a patient. The documents are provided to a federated document repository and registry with its corresponding metadata. The document registry then creates a longitudinal record of information about a patient within a given clinical affinity domain. This information is then used by the other healthcare organizations to query and retrieve documents. This profile is based upon ebXML Registry standards, SOAP, HTTP and SMTP. It describes the configuration of an ebXML Registry in sufficient detail to support Cross Enterprise Document Sharing. The main actors and the transactions among them are depicted in Figure 3.1.

The Document Source Actor is the producer and publisher of documents. Hospital Information Systems may be an example for this actor. It is responsible for sending documents and the metadata of the documents to a Document Repository Actor. The metadata that can be used in XDS profile is fixed. The followings are some sample metadata information that can be used in XDS.

- authorInstitution: Represents a specific healthcare facility under which the human and/or machines authored the document.

- authorPerson: Represents the humans and/or machines that authored the document within the authorInstitution.

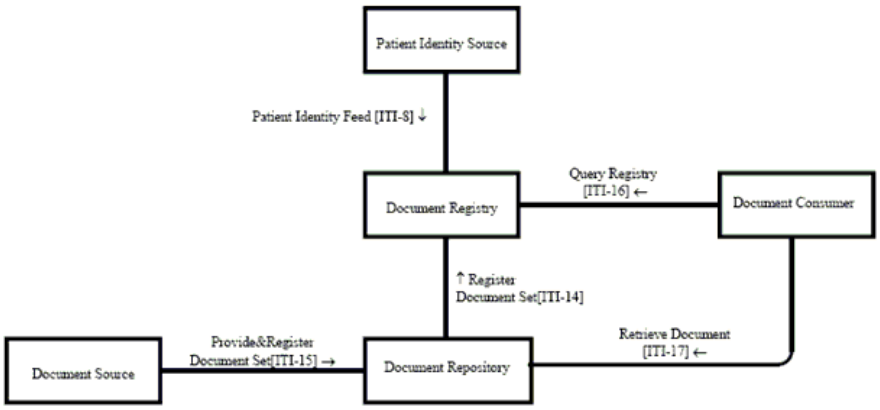

Figure 3.1: IHE XDS Actors and Transactions

- classCode: The code specifying the particular kind of document (e.g. Prescription, Discharge Summary, Report).
- confidentialityCode: The code specifying the level of confidentiality of the XDS Document (Use of this attribute is illustrated in the next sections).
- creationTime: Represents the time the author created the document in the Document Source.
- patientId: The patientId represents the subject of care medical record identifier as selected by the Document Source.

Metadata information is used to describe and query the documents in registry. From the privacy perspective, the privacy and access control policies and rules for the documents can be based on these metadata attributes. In my implementation, the use of authorInstitution, classCode, confidentialityCode and patientId is illustrated in the following sections. The Document Repository is responsible from storing the documents as well as registering them to the appropriate Document Registry. When a Document Source actor sends the documents and metadata with the "Provide and Register Document Set" transaction, the Document Repository stores the document, assigns a unique URI to document,

add this URI to metadata and send the metadata to the Registry actor. Usually Document Repository and Document Source actors are located in the same system. For instance, a hospital can implement Document Source and Document Repository actors and can store its documents in its internal. The Document Registry Actor maintains metadata about each registered document in a document entry. This includes a link to the Document in the Repository where it is stored. The Document Registry responds to queries from Document Consumer actors about documents meeting specific criteria and returns the URI of the document where it is stored. Then Document Consumer uses this URI to retrieve the document from the Document Repository.

3.1.2 IHE Cross User Authentication Profile

IHE Cross Enterprise User Authentication(XUA) is a profile candidate which aims to provide the user identity in transactions that cross enterprise boundaries. The XUA whitepaper [27] provide some first considerations about the profile. The SAML profiles are considered for XUA specification. In this thesis, we describe how to implement this profile with SAML ECP profile for XDS transactions.

3.1.3 IHE Audit Trailing and Node Authentication (ATNA)

The Audit Trail and Node Authentication (ATNA) Integration Profile establishes security measures which, together with the Security Policy and Procedures of the enterprise, provide patient information confidentiality, data integrity and user accountability. The goals of the Audit Trail and Node Authentication Integration Profile are:

- User Accountability (Audit Trail)
- Access Control
- Centralized Audit Record Repository

- Protected Health Information (PHI) Data Integrity

ATNA profile has three key features to achieve the above goals. One of these is the authentication of the user to the node. Although IHE does not restrict any technology for the user authentication, it recommends Enterprise User Authentication profile as the mechanism for authentication to nodes. Another feature is the audit record generation in which related actors generate records about PHI and send these to a repository. As a result, this feature enables monitoring of events related with PHI and detecting inappropriate activity. The final feature is the node authentication between nodes during their communication. This feature is not related with the user authentication, it is mostly about the authorization issues between actors or application parts (nodes) while transferring PHI data between them.

ATNA uses auditing as part of a security and privacy process needed independently of the access control and authentication methods. Auditing is needed for situations where the people involved are generally trustworthy and need a wide range of flexibility to respond rapidly to changing situations which is the typical healthcare provider environment. Auditing tracks what takes place, and the people involved know that their actions are being audited. This means that the audit records must capture event descriptions for the entire process, not just for individual components that correspond to individual IHE actors. The IHE audit trail is the first of several profiles that correspond to different forms of access control and authentication.

The "Security Audit and Access Accountability Message XML Data Definitions for Healthcare Applications" (RFC-3381) [26] defines an XML schema for reporting events that are relevant to security and privacy auditing. It was defined in cooperation with the ASTM, HL7, and DICOM standards organizations and the NEMA/COCIR/JIRA Security and Privacy Committee. The IHE recommends the use of the RFC-3381 format, and recommends reporting only events that it can describe.

The Audit Trail and Node Authentication Integration Profile specifies the use of Reliable Syslog Cooked Profile (RFC-3195) as the mechanism for logging audit record messages to the central audit record repository. It also permits the use of BSD Syslog (RFC-3164).

3.2 OASIS Security Assertion Markup Language(SAML)

SAML defines and maintains a standard XML-based framework for creating and exchanging authentication and authorization information. SAML uses the approach of expressing assertions about a subject in a portable fashion that other applications across system domain boundaries can trust. The three main actors in SAML are Service Clients, Service Provider and Identity Provider. The Identity Providers are the actors which have the identity information of users and assert these identities to Service Providers when it is asked.

SAML consists of a number of building-block components that, when put together, allow a number of use cases to be supported. Primarily the components permit transfer of identity, authentication, and authorization information to be exchanged between autonomous organizations. The components of SAML and their hierarchic view is illustrated at Figure 3.2.

Assertions: SAML allows one actor to claim some information about another actor and this is called assertion. SAML defines three kinds of statements that can be carried by assertions:

- Authentication Statement: These are issued by the party that successfully authenticated the user. They define who issued the assertion, the authenticated subject, validity period, plus other authentication related information (e.g. John Doe is authenticated with username/password method at 10:00 am to www.example.com).

- Attribute Statement: These contain specific details about the user (e.g. The User has a role of General Practitioner).

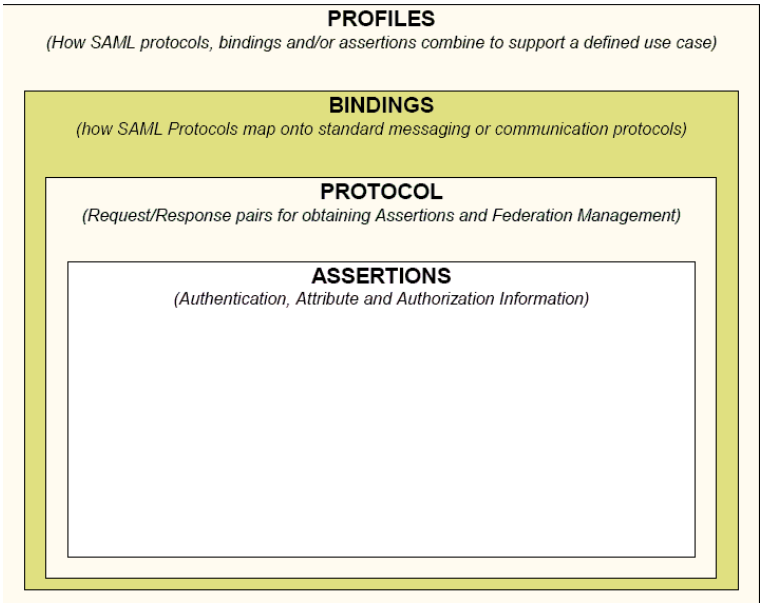

Figure 3.2: SAML Components

- Authorization decision statements: These identify what the user is entitled to do (e.g. John Doe is permitted to view Medical Summary of Mary Brown).

Protocols: SAML defines a number of request/response protocols, which are encoded in an XML schema as a set of request-response pairs. The protocols defined are:

- Assertion Query and Request Protocol: Defines a set of queries by which existing SAML assertions may be obtained. The query can be on the basis of a reference, subject, or the statement type.

- Authentication Request Protocol: Defines a protocol by which a Service Provider or Principal can request authentication statements from an Identity Provider.

- Artifact Resolution Protocol: Provides a mechanism by which protocol messages may be passed by reference using a small, fixed-length value

called an artifact.

- Name Identifier Management Protocol: Sometimes different Service Providers or Identity providers can use different names for Principals. This protocol provides mechanisms to change the value or format of the name of a Principal in an actor.

- Single Logout Protocol: Defines a request that allows simultaneous logout of all sessions in different providers associated by a Principal.

- Name Identifier Mapping Protocol: Provides a mechanism to programmatically map one SAML name identifier into another, subject to appropriate policy controls.

Bindings: The bindings describe the details exactly how the SAML protocol maps onto the transport protocols. For instance, the SAML specification provides a binding of how SAML request/responses are carried with SOAP exchange messages. The bindings defined are: SAML SOAP Binding, Reverse SOAP (PAOS) Binding, HTTP Redirect Binding, HTTP Post Binding, HTTP Artifact Binding, and SAML URI Binding.

Profiles: The SAML profile defines how assertions, protocols and bindings can be combined for interoperability in particular usage scenarios. Current SAML profiles are as follows:

- Web Browser SSO Profile: Defines a mechanism for single sign-on by unmodified web browsers to multiple Service Providers using the Authentication Request protocol in combination with the HTTP Redirect, POST, and Artifact bindings.

- Enhanced Client and Proxy (ECP) Profile: Defines a profile of the Authentication Request protocol in conjunction with the Reverse-SOAP and SOAP bindings suited to clients or gateway devices with knowledge of one or more Identity Providers.

- Identity Provider Discovery Profile: Defines one possible mechanism for a set of cooperating Identity and Service Providers to obtain the Identity Providers used by a Principal.

- Single Logout Profile: A profile of the SAML Single Logout protocol is defined. Defines how SOAP, HTTP Redirect, HTTP POST and HTTP Artifact bindings may be used.

- Name Identifier Management Profile: Defines how the Name Identifier Management protocol may be used with SOAP, HTTP Redirect, HTTP POST and HTTP Artifact bindings.

- Artifact Resolution Profile: Defines how the Artifact Resolution protocol uses a synchronous binding, for example the SOAP binding.

- Assertion Query/Request Profile: Defines how the SAML query protocols (used for obtaining SAML assertions) use a synchronous binding such as the SOAP binding.

- Name Identifier Mapping Profile: Defines how the Name Identifier Mapping protocol uses a synchronous binding such as the SOAP binding.

Two other SAML components can be used in building a system:

- Metadata: Metadata defines how to express and share configuration information between two communicating entities. For instance, an entity's support for given SAML bindings, identifier information, and PKI information can be defined. Metadata is defined by an XML Schema. The location of Metadata is defined using DNS records.

- Authentication Context: In a number of situations the Service Provider may wish to have additional information in determining the authenticity and confidence they have in the information within an assertion. Authentication Context permits the augmentation of Assertions with additional

information pertaining to the authentication of the Principal at the Identity Provider. For instance, details of multi-factor authentication can be included.

It should be noted that SAML Assertions provide a means to distribute security-related information as input to Access Control decisions such as when and how a user authenticated or what attributes are used in deciding if a request should be allowed. However SAML does not specify how this information should be used or how access control policies should be addressed. OASIS has specified another standard, namely, eXtensible Access Control Markup Language (XACML) for this purpose.

3.3 OASIS Extensible Access Control Markup Language (XACML)

The eXtensible Access Control Markup Language (XACML) is an OASIS Standard that defines the syntax and semantics of a language for expressing and evaluating access control policies. It provides an XML schema for a general policy language which is used to protect any kind of resource and make access decisions over these resources. XACML standard not only gives the model of the policy language, but also proposes a processing environment model to manage the policies and to conclude the access decisions.

3.3.1 XACML Processing Environment

The XACML profile specifies five main actors to handle access decisions: Policy Enforcement Point (PEP), Policy Administration Point (PAP), Policy Decision Point (PDP), Policy Information Point (PIP) and a context handler.

- Policy Administration Point (PAP): PAP is the repository for the policies and provides the policies to the Policy Decision Point (PDP).
- Policy Enforcement Point (PEP): PEP is actually the interface of the whole environment to the outside world. It receives the access requests

and evaluates them with the help of the other actors and permits or denies the access to the resource.

- Policy Decision Point (PDP): PDP is the main decision point for the access requests. It collects all the necessary information from other actors and concludes a decision.

- Policy Information Point (PIP): PIP is the point where the necessary attributes for the policy evaluation are retrieved from several external or internal actors. The attributes can be retrieved from the resource to be accessed, environment (e.g., time), subjects etc.

3.3.2 XACML Language Model

The language model is composed of three main components which are Rules, Policies and PolicySets.

- Rule: A Rule element defines the target elements to which the rule is applied and gives conditions to apply the rule. It has three components namely, target, effect and condition. The target element defines the resources, subjects, actions and the environment to which the rule is applied. As an example, assume that a physician is willing to access a patient's clinical summary data. If an XACML rule is defined for such a case, the resource target element can be the clinical summary data type, target action can be the retrieval of this data and the environment can restrict the access time. To apply the rule, all these entities should match the defined subjects, resources, actions and environment in the request context. The 'Condition' element gives the conditions to apply the rule. Continuing with the example, the condition can be that the physician's ID number accessing the patient's record should match the physician's ID number on the clinical summary data. 'Effect' is the consequence of the rule as either 'permit' or 'deny'.

- Policy: Policies are the set of rules which are combined with some algorithms. The algorithms used are called Rule-combining algorithms. For example, “Permit Override” algorithm allows the Policy to evaluate to “Permit” if any rule in the policy evaluates to “Permit”. If a rule evaluates to “Deny” and all others are “Not Applicable”, the result is “Deny”. If all the rules are “Not Applicable” then the policy is “Not Applicable”. Policies have also ’target’ elements which give the subjects, resources, actions and environment that policy is applied. For example, a policy’s subject can be a user role to indicate that the policy restricts the access rights related with that user role. Another component is the obligations which define necessary actions if the policy is evaluated to ’Permit’ before giving access to resource. For example, a possible obligation is to send an email to the patient when his clinical summary data is accessed.

- PolicySet: A set of policies compose the policy sets by policy-combining algorithm as in the policy. It has also ’target’ and ’obligations’ components with the same semantics.

3.3.3 XACML Context

The request and response context used by the Policy Decision Point (PDP) are also defined in the standard. This context is actually the interface to the application environment to insulate the core XACML language. Therefore, applications can use other representations like SAML which is the most suitable one for the attributes. Then the Policy Enforcement Point (PEP) applications convert these attribute representations to the XACML context attributes.

CHAPTER 4

A PRIVACY INFRASTRUCTURE FOR XDS

In this chapter, the proposed privacy infsratructure with cross user authentication, access control and auditing functionalities for XDS affinity domain are described. The chapter is organized as follows: Section 4.1 gives the high level overview of the overall system and a scenario which illustrates the implementation, Sction 4.2 describes the details of cross user authentication process and discussions, Section 4.3 provides details and discussions on acess control mechanism, and finally Section 4.4 provides the details of audit mechanism.

4.1 Overal Privacy Infrastructure

The implementation is realized through a scenario which is based on an XDS Affinity Domain where the "Electronic Health Record (EHR)" of a patient is shared between two healthcare institutes. The implementation concentrates on the "XDS Retrieve Transaction" and provides the "Cross User Authentication" and "Patient Consent" based authorization services on this transaction. For "Cross User Authentication", SAML ECP profile is implemented. The authorization service is implemented based on Role Base Access Control (RBAC) model using the OASIS Extensible Access Control Markup Language (XACML) standard. Furthermore, IHE ATNA Audit Record Repository actor is imple-

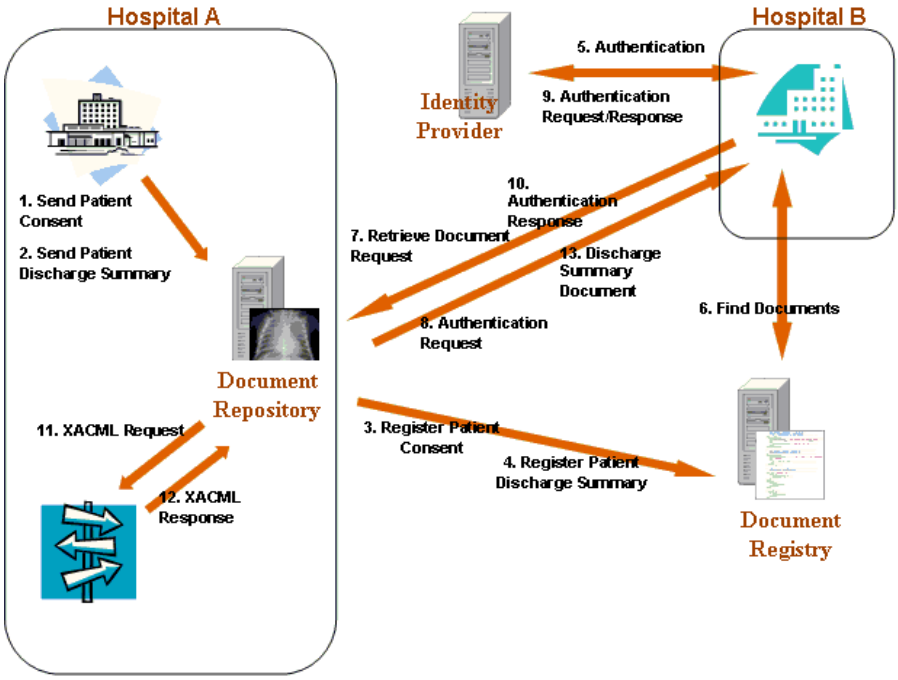

Figure 4.1: The Implementation Scenario

mented to record logs for events in XDS affinity domain.

4.1.1 Implementation Scenario

The scenario is illustrated in Figure 4.1 and the steps are described in Figure 4.2.

In step 0.b, the patient consent is generated with the help of a web base consent editor tool. The tool produces an XACML policy which corresponds to consent rules. The policy is based on RBAC model where the healthcare professional roles and classification of documents in terms of sensitivity have been already specified in the XDS affinity domain. The Consent Editor is designed in three modes. The basic mode of the Consent Editor provides some basic choices for the patient. The choices are defined by simple sentences. The editor binds these sentences to the appropriate rules which are defined as configurations of the editor. The advance mode provides an interface to match the user roles and

Step	Scenario	Interoperability Mechanism
0.a	A patient with cardiovascular problems experienced a heart problem when he is abroad.	
0.b	Hospital A creates a "Patient Discharge Summary" in and the Patient Consent and wants to register these document to EHR Registry/ Repository.	
1	The Hospital A stores the "Patient Consent" to the EHR Repository	IHE XDS
2	The EHR Repository registers the "Patient Consent" document to the EHR Registry.	IHE XDS
3	Hospital B stores the "Patient Discharge Summary" to the EHR Repository	IHE XDS
4	The EHR Repository registers the "Patient Discharge Summary" document to the EHR Registry.	IHE XDS
5	After he returned to his home his primary physician at Hospital B, Dr. John Doe, wants to access his Discharge Summary. He authenticates himself to the Local system, and this authentication informations is sent to the Identity Provider.	
6	John Doe wishes to see the available Electronic Healthcare Records of the Patient from the EHR Registry/Repository. Therefore he queries the available EHRs of the patient from the EHR Registry.	IHE XDS
7	Using the document link provided by the EHR Registry, the doctor requests to retrive the "Patient Discarge Summary" from the EHR Repository.	IHE XDS
8.a	The EHR Repository, which is an internal repository of Hospital A, sends an Authentication request to the Hospital B, specifying the Identity provider list any of which should be contacted to receive the authentication assertion.	SAML 2.0 ECP Profile
9.a	Hospital B forwards the Athentication request to the Identity Provider. Since the authentication information of the doctor in Hospital B previously sent to the Identity Provider, it creates an authentication response as a signed assertion and send it to the Hospital B.	SAML 2.0 ECP Profile
10.a	Hospital B sends the Authentication Response to the EHR Repository	SAML 2.0 ECP Profile
8.b	The EHR Repository requests some further information like functional role of the user.	SAML Attribute Profiles
9.b	The Hospital B forwards the request to Identity Provider and retrieve the information as an attribute statement.	SAML Attribute Profiles
10.b	Hospital B sends the Attribute Statements to the EHR Repository	SAML Attribute Profiles
11	The Policy Enforcement Point deployed in the EHR Repository, constructs a XACML Request from the information that it retrieves and send to the Policy Decision Point.	XACML 2.0, XACML 2.0 RBAC Profile
12	The Policy Decision Point executes the XACML request on the Patient Consent documnts and concludes that the doctor is athorized to see the discharge summary document.	XACML 2.0, XACML 2.0 RBAC Profile
13	The EHR Repository returns the Patient Discharge Summary Document to the Hospital B	IHE XDS
14	All transactions are secure and logged at the **Audit Record Repository.**	IHE ATNA

Figure 4.2: Scenario Steps

classification of documents. In addition, the patients can set some restrictions, for example time ranges for access and mail obligation (e.g. when the document is accessed a mail must be sent to the patient).

After the policy is obtained, it is sent to XDS Repository by the institute. While sending the consent through the "XDS Provide and Register Document" transaction, some of the attributes in the metadata should be set. For example, class code is set as Consent which marks the document as consent. After sending

the consent, the medical documents are also sent. In their metadata, confidentialityCode entry specifies their sensitivity (e.g. General Clinical Information).

4.1.2 Overview of Components

In this section, an overview of components that are implemented within the scope of this thesis are given. More details about the components and transactions realized among them are given in other sections.

- *Enhanced Client:* This is the component for XDS Document Consumer actors which provides session management between the actor and the Identity Provider and enables the XDS consumers to handle SAML ECP transactions. The Session Manager software provides a service which establishes a session on Identity Provider. The component is configurable with SAML Metadata documents.

- *Service Provider:* This is the component which can be used by XDS Document Repositories to cross authenticate users and apply access control. The subcomponents of the Service Provider are SAML Single Sign-on Handler, SAML Attribute Handler and Policy Enforcement Point. The SAML Single Sign-on Handler handles the SAML ECP transactions executed between Enhanced Client to authenticate the user of the Enhanced Client actor. SAML Attribute Handler handles the attribute queries from Identity Provider which is based on SAML Attribute Query Profile. The Policy Enforcement Point provides a plug-in to XDS Repository actor. It stores XACML based patient consents in to the repository and when a document is requested it finds the related consent and force the policy execution for access control. The Policy Enforcement Point communicates with Policy Decision Point to get the access decision.

- *Identity Provider:* The Identity Provider actor manages the identity, authentication and attributes information of users. The Attribute Authority

subcomponent manages the further information about the users and provides this information as Attribute Statements as a response to SAML Attribute Queries sent by Service providers. The Identity Provider also stores the session information for the user authentications and provides this information as Authentication Statements.

- *Audit Record Repository:* This is the component which implements the IHE ATNA profile and provides a repository for the audit records. The component also provides a user interface to view the audit records.

4.2 Cross User Authentication and Trust

In this chapter the proposed trust model and the cross user authentication mechanism based on SAML Enhance Client Proxy profile are described.

4.2.1 Trust Model for XDS Affinity Domain

Both in SAML and WS-Trust [9] specifications, the trust model depend on the delegation of trust to the intermediary trust brokers. These intermediary actors are called X-Identity Provider in IHE XUA profile, Identity Provider (IDP) in SAML and Security Token Service (STS) in WS-Trust model. In this way, the providers of services that need user authentication (users identity) only need to trust the claims of these trusted entities.

Several identity management models can be constructed by using these intermediaries. However, considering the IHE affinity domain concept, the model would be as given in Figure 4.3.

In this model, any service provider should have trust relationships with all X-Identity Provider actors. This trust should be in two ways; that is, any service provider should trust the claims of the X-Identity Provider that is located in the affinity domain and any X-Identity Provider should ensure that the entity which the claim is sent to is one of the authorized service providers in the affinity domain. On the other hand, one X-Identity provider can serve claims of several

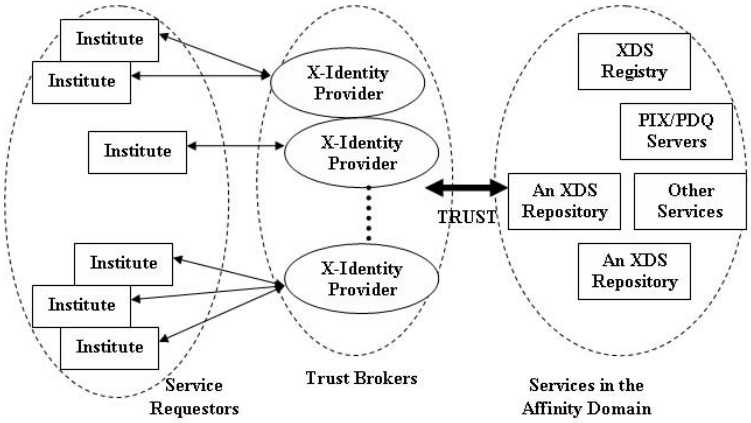

Figure 4.3: A Trust Model for IHE Affinity Domain

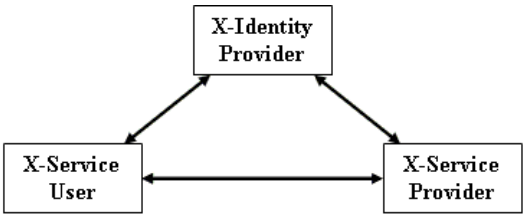

Figure 4.4: XUA Actors

institutes. In this model, using a single X-Identity Provider for the whole affinity domain can also be considered.

In this abstract model, any service (XUA enabled) request within the affinity domain is accompanied by some XUA transactions including the three XUA actors; namely X-Service Provider, X-Identity Provider and X-Service User. All these three actors communicate with each other to share the user identity. The arrows in Figure 4.4 represent the communication paths between these actors. The order of communication paths is important and should be restricted for interoperability. In fact, one of the important functionality of SAML Profiles is restricting the message flow to certain patterns. WS-Trust specification also mentions such future profiles to give restrictions over communication flows.

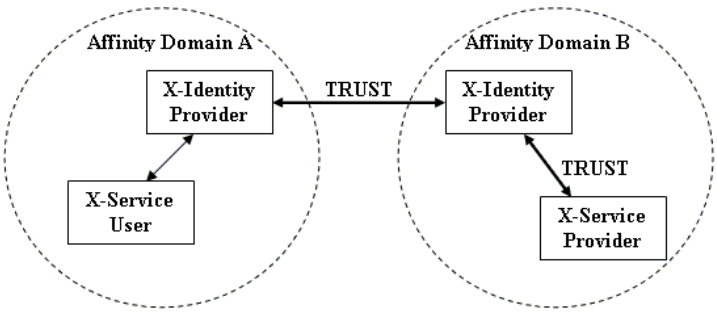

Figure 4.5: Trust in Federation of Affinity Domains

XUA profile provides two process flows which are "Pre-Generated Assertions" and "Post-Generated Assertions". In order to use "Pre-Generated Assertions", clients should be aware of the service provider preferences and restrictions. The X-Service User gets the assertion according to the known X-Service Provider preferences from the X-Identity Provider and sends this assertion together with service request to X-Service Provider. In the Post-Generated Assertions case, after the request of the X-Service User, the X-Service Provider requests an assertion by defining its preferences as a response.

4.2.1.1 Trust Model for Federated Affinity Domains

When we also consider the federation of affinity domains, the model given in the Figure 4.4 can be extended as shown in Figure 4.5. In this case, X-Service Provider in the Affinity Domain B does not have a direct trust relationship with the X-Identity Provider in the Affinity Domain A. Therefore, claims of this identity provider for X-Service User are not acceptable by X-Service provider. We propose to extend the trust chain to include the Identity Providers in both of the affinity domains. In this way it becomes possible to manage the identity of the user across affinity domains. SAML and WS-Trust specifications both mention and supplement such scenarios. In fact, the trust between the X-Identity Providers does not need to be a direct trust.

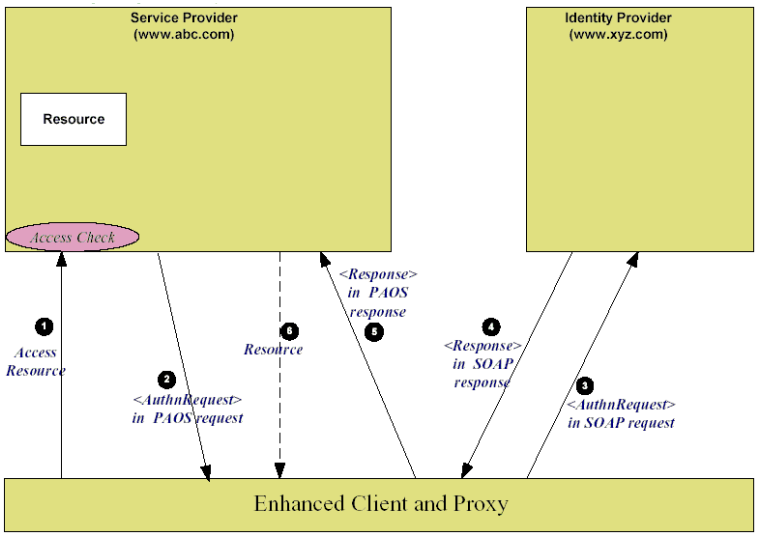

Figure 4.6: ECP Message Flow

4.2.2 Cross User Authentication with SAML ECP Profile

In this section an example is given describing the whole cycle of ECP profile. This section also includes discussions on ECP profile capabilities and limitations from IHE viewpoint. Web Single Sign-On profile is also included in these discussions.

"An enhanced client or proxy (ECP) is a system entity that knows how to contact an appropriate identity provider, possibly in a context-dependent fashion, and also supports the Reverse SOAP (PAOS) binding" [28]. From the definition it is clear that ECP communicates directly with its Identity Provider (IDP). The whole message flow is shown in the Figure 4.6 [29].

An example message flow from the implementation is given in the following sections to demonstrate some of the details of the ECP Profile.

4.2.2.1 SAML AuthnRequest

The Figure 4.7 shows the HTTP GET request for the XDS Retrieve transaction from an ECP actor to retrieve a EHR Document from the Document Repository.

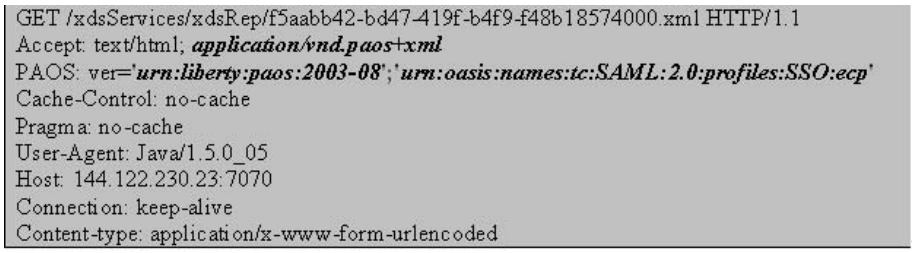

Figure 4.7: XDS Retrieve HTTP Request

The only change in this message is the PAOS header which shows that the ECP actor has implemented the ECP Profile and use the PAOS binding.

When the X-Service Provider (SP) receives this request, it checks the PAOS header to ensure that the requester supports SAML ECP Profile. After verification, SP constructs a SAML AuthnRequest message from its configuration and preferences. We define the SP's preferences through some configuration files. SAML provides SAML Metadata specification [30] in this respect. However the API, openSAML 2.0, that is used for implementation has not implemented Metadata section yet so simple configuration xml files are used. A brief description of SAML Metadata specification and some examples are given in the following sections.

The objective of SAML AuthnRequest element is to request an authentication statement from a trusted IDP about a user. In this respect, it is suitable for XDS transactions. SP can declare its preferences and restrictions over the authentication and the response that it receives. The Figure 4.8 shows an example SAML AuthnRequest element.

The most important element in this AuthnRequest is the RequestedAuthnContext element. It defines the SP's restrictions for the user authentication in the Service User side. In our example, we state that authentication method must be the method defined with the unique URI urn:oasis:names:tc:SAML:2.0:ac:classes:Password. SAML has defined such methods in the SAML Authentication Context specification [31]. It also allows declaring such contexts

```
<samlp:AuthnRequest
        xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
        AssertionConsumerServiceURL="https://144.122.230.23:8443/xdsServices/xdsRep/AssertionConsumer"
        ForceAuthn="false"
        ID="_2zC8NK7kjZNvqaT"
        IsPassive="true"
        IssueInstant="2006-10-03T06:29:03.450Z"
        ProviderName="SRDC-XDS" Version="2.0">
        <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" SPProviderID="sp"/>
        <saml:Conditions xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
        NotOnOrAfter="2006-10-03T06:32:03.450Z"/>
        <samlp:RequestedAuthnContext Comparison="exact">
                <saml:AuthnContextClassRef
                xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
                        urn:oasis:names:tc:SAML:2.0:ac:classes:Password
                </saml:AuthnContextClassRef>
        </samlp:RequestedAuthnContext>
        <samlp:Scoping>
                <samlp:IDPList>
                        <samlp:IDPEntryLoc="http://144.122.230.23:9091/identityprovider/IdentityProvider"
Name="SRDC-Care2x" ProviderID="idp"/>
                        <samlp:GetComplete>/TrustedIDPList.xml</samlp:GetComplete>
                </samlp:IDPList>
        </samlp:Scoping>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                .....
        </ds:Signature>
</samlp:AuthnRequest>
```

Figure 4.8: SAML AuthnRequest

by giving reference to xml schemas.

SAML Scoping element gives the information about IDPs which the SP trusts. In our example scenario, this is all the Identity providers in the same affinity domain of SP.

In Figure 4.8, the signature of SP is shown which the IDP uses to authenticate and verify the integrity of the message. ECP profile does not mandate the signature for AuthnRequest element but it is strongly recommended. However, it states that IDP must verify any AssertionConsumerServiceURL which must be the real endpoint of the SP whose identifier is given in the Issuer element. In our configuration, IDP has a list of trusted SPs with their IDs, assertion consumer URLs and certificates. This information is used to authenticate SP, check the AssertionConsumerServiceURL attribute and verify the integrity of the message.

AuthRequest element is defined in the SAML Core specification [32]. ECP Profile sets the restrictions about the message binding and set some more re-

strictions over the XML structure as in the other SAML profiles. The sample message with PAOS binding is shown in Figure 4.9. The message has two required headers; paos:Request and ecp:Request. These headers are for the use of ECP and removed by ECP before forwarding the message to IDP. The messageID attribute of paos:Request header is used to correlate the message with the SOAP response. Nevertheless, since the AuthnRequest element has also a unique id which is used for the same purposes, this attribute in the paos:Request header is optional. In the implementation this attribute is set by the SP and used as session ID when waiting for the authentication response. By using this attribute, the message correlation in SOAP header level is provided.

SP sends the message including the AuthnRequest to the ECP. ECP process the headers, remove them and forwards the message to its IDP with SAML SOAP binding. As described in the Trust Model section, we assume that each ECP has one IDP which serves assertion about the users in the system that ECP is working. ECP profile does not specify anything about how to choose the IDP. Only the selected IDP should be trusted by the SP. As mentioned before, the list of trusted IDPs are given in the ecp:Request header. In addition to these, the value of responseConsumerURL attribute of paos:Request header should be stored. This value gives SP's endpoint URL of the services which processes the response from the ECP.

The ECP profile recommends the use of SSL 3.0 [33] or TLS 1.0 [34] in order to maintain confidentiality and integrity of the whole message. In fact, the integrity of the SAML elements inside the message like AuthnRequest element is protected by signatures however the SOAP headers also should be protected. This is already mentioned in the whitepaper of XUA profile [27] by recommending the use of IHE ATNA profile [22] together with XUA. IHE ATNA uses the TLS 1.0 to provide node authentication and provide confidentiality and integrity of the whole communication line between the nodes. In this way, the SOAP headers are also protected as requested in the ECP profile.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <SOAP-ENV:Header>
                <paos:Request xmlns:paos="urn:liberty:paos:2003-08"
                        SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
                        SOAP-ENV:mustUnderstand="1" messageID="7J6iylkrkM6KB"
responseConsumerURL="https://144.122.230.23:8443/xdsServices/xdsRep/AssertionConsumer"
                        service="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"/>
                <ecp:Request xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
                IsPassive="1" ProviderName="SRDC-XDS"
                        SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
                        SOAP-ENV:mustUnderstand="1">
                        <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
SPProviderID="sp"/>
                        <samlp:IDPList xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
                                <samlp:IDPEntry Loc="http://144.122.230.23:9091/identity-
provider/IdentityProvider" Name="SRDC-Care2x" ProviderID="idp"/>
                                <samlp:GetComplete>/TrustedIDPList.xml</samlp:GetComplete>
                        </samlp:IDPList>
                </ecp:Request>
        </SOAP-ENV:Header>
        <SOAP-ENV:Body>
                <samlp:AuthnRequest>
                        ....
                </samlp:AuthnRequest>
        </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 4.9: SAML AuthnRequest Message PAOS Binding

4.2.2.2 Identification of Principal by IDP

The IDP should identify the user (subject) for whom the SP wants authentication statement. The ECP Profile optionally allows the use of SAML Subject element inside the AuthnRequest element. By using Subject element, the SP can state the principal for whom it requests authentication statement (assertions). However, in order to do this it needs to identify the principal from the service request. SAML does not specify anything for this; neither IHE has such specifications for its services. As seen from the Figure 4.8, such a Subject element is not included. In this case SAML Core [32] specification states that presenter of the message is assumed to be the subject. Nevertheless, in both cases, IDP should identify the principal. This identification is not only discovering an id or a name of the subject but establishing a security context with the principal. This step is mentioned but not included in the ECP profile scope. It only states that IDP must establish the identity of principal by any means; either it may

start a new act of authentication or may reuse existing authentication session.

This issue is very critical in an IHE affinity domain. If X-Identity Providers are planned to be individual external entities as in our model given (because self assertion scenarios can not represent real life as discussed in open issues in the whitepaper [27]), the methodology and the interface between IDP and ECP while establishing the security context should be clearly specified. XUA whitepaper [27] has also mentioned this relationship between the corresponding actors User Authentication Provider (which is assumed to be located in ECP), X-Assertion Provider (XUA terminology for Identity Provider). Nevertheless, the profile also states that this relationship is out of scope that is how X-Assertion Provider gets the authentication information to create an assertion is left to the implementations.

In the light of these discussions, how the implementation handles this step is as follows. In the implementation framework, when a user is authenticated to the web interface (simple user-password authentication) for the Document Consumer actor which is actually our ECP, the authentication information is sent to the IDP. This transformation is simple HTTP Post and the content of the transaction is not bound to any standard. When IDP receives the authentication information it generates a session for the user (stores authentication info) and sends a cookie to establish the security context. The cookie is used by the ECP for the transactions between the IDP so that the IDP can identify the principal and generates the assertion from the authentication information in the session. The SAML has provided an Implementation Guideline [35] which discusses how cookies can be used in session state maintenance and security context establishment.

In summary, standard mechanisms need to be specified for the relationship between IDP and ECP to exchange authentication information of user. SAML Assertion Query Request Protocols can be easily used for this purpose with some context management mechanisms like cookies. However, some one way protocol may be needed in which authentication statement is directly sent without a

query or a request. Furthermore, WS-SecureConversation specification [10] has the main objective of establishing security context and may be used if the web service transactions are used. More details are provided on this issue in the following sections.

4.2.2.3 IDP Response

When IDP identifies the principal, we can assume that any information that is needed for the response is ready. The next step for IDP is to check if the authentication preferences of SP given in RequestedAuthnContext element are satisfied by the ECP when authenticating the user to the ECP's system. From this authentication information, IDP generates an AuthenticationStatement. The Figure 4.10 shows the SAML Response element including the AuthenticationStatement which gives the authentication instant and method.

SAML Response element has an InResponseTo attribute which is actually the ID of AuthnRequest sent by SP. This attribute provides message correlation. The Status element shows the status of the response; urn:oasis:names:tc:SAML:2.0:status:Success means IDP has the authentication information of the user which is suitable for the SP's preferences. This attribute can take other values to describe the problems that can occur while checking user authentication, authentication of SP, signature verifications, processing of the message. Some of these values which is used in the implementation are as follows:

- urn:oasis:names:tc:SAML:2.0:status:VersionMismatch
- urn:oasis:names:tc:SAML:2.0:status:RequestDenied
- urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext
- urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal

ECP profile allows more than one SAML Assertion element inside the Response. In our scenario, only one assertion is used to give the authentication statement. The Assertion must include a Subject which gives a name identifier

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" ID="_4o9XZa7wzfwsjKu"
InResponseTo="_2zC8NK7kjZNvqaT" IssueInstant="2006-10-03T06:30:05.778Z" Version="2.0">
        <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" SPProviderID="idp"/>
        <samlp:Status>
                <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
        </samlp:Status>
        <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_1Nt7I5qek4IWxRc"
IssueInstant="2006-10-03T06:30:05.778Z" Version="2.0">
                <saml:Issuer SPProviderID="idp"/>
                <saml:Subject>
                        <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">doe</saml:NameID>
                        <saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                                <saml:SubjectConfirmationData
InResponseTo="_2zC8NK7kjZNvqaT" NotOnOrAfter="2006-10-03T06:33:05.778Z"
Recipient="https://144.122.230.23:8443/xdsServices/xdsRep/AssertionConsumer"/>
                        </saml:SubjectConfirmation>
                </saml:Subject>
                <saml:Conditions>
                        <saml:AudienceRestriction>
                                <saml:Audience>sp</saml:Audience>
                        </saml:AudienceRestriction>
                </saml:Conditions>
                <saml:AuthnStatement AuthnInstant="2006-10-03T06:15:05.778Z"
SessionNotOnOrAfter="2006-10-03T07:15:05.778Z">
                        <saml:SubjectLocality Address="127.0.0.1"/>
                        <saml:AuthnContext>
                                <saml:AuthnContextClassRef>
                                        urn:oasis:names:tc:SAML:2.0:ac:classes:Password
                                </saml:AuthnContextClassRef>
                        </saml:AuthnContext>
                </saml:AuthnStatement>
                <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                        ...
                </ds:Signature>
        </saml:Assertion>
</samlp:Response>
```

Figure 4.10: SAML Response

for the subject of the assertion. In the implementation, user ID of the user in the ECP's system is used as a name identifier. In real life, if we consider the health domain, this should be a health professional identifier which is unique for the affinity domain. SAML provides Name Identifier Mapping Profile which proposes a solution when two parties (SP and IDP) do not use the same identifiers for the user. WS-Federation has also defined some use cases for this purpose.

The Subject element must include a SubjectConfirmation element. The element is used by the relying party (SP in our case) to confirm that the request or message came from a system entity (ECP in our case) that is associated with the subject of the assertion [36].

The Method attribute of SubjectConfirmation element gives an identifier of

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <SOAP-ENV:Header>
                <ecp:Response
                        xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
                        AssertionConsumerServiceURL="https://144.122.230.23:8443/xdsServices/xds
Rep/AssertionConsumer"
                        SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
                        SOAP-ENV:mustUnderstand="1"/>
        </SOAP-ENV:Header>
        <SOAP-ENV:Body>
                <samlp:Response>
                        ...
                </samlp:Response
        </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 4.11: IDP Response with SOAP Binding

the method which SP must use to confirm that the subject of the assertion is actual subject of the request. SAML has defined three such methods which are used in the SAML profiles.

ECP Profile mandates the use of Bearer method as it is used in the example shown in Figure 4.10. The SubjectConfirmationData element states that the bearer (carrier) of this assertion can be confirmed to be the real subject only if the assertion is delivered in a message sent to https://144.122.230.23:8443/xdsServices/xdsRep/AssertionConsumer before 2006-10-03T06:33:05.778Z.

The other methods can optionally be used in ECP profile. Holder-of-Key method is more secure. In this method the IDP put some information about the key of the subject into the SubjectConfirmationData element. This information can be the name of the key or the whole key data (e.g X509 Public certificate). The requesting party (ECP in our case) signs the assertion with the private key of the user before sending it to the relying party (SP in our case). SP use the information inside the SubjectConfirmationData (it may be the whole certificate and can be directly use as a public key for signature verification) to find the key and verify the signature of the subject by using this key. If verification is successful, subject is assumed to be confirmed.

Finally IDP must sign all assertions inside the Response. It may also sign the Response element. These signatures are verified by SP in order to be sure

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <SOAP-ENV:Header>
                <paos:Response
                        xmlns:paos="urn:liberty:paos:2003-08"
                        SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
                        SOAP-ENV:mustUnderstand="1"
                        refToMessageID="7J6iyIkrkM6KB "/>
        </SOAP-ENV:Header>
        <SOAP-ENV:Body>
                <samlp:Response>
                        ...
                </samlp:Response>
        </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 4.12: ECP forwards the Response with PAOS binding

about the integrity of the Response. IDP sends the Response to the ECP with SOAP binding. As seen from Figure 4.11, ecp:Response header must be used in the SOAP message. The AssertionConsumerServiceURL attribute is set by IDP by using the value of AssertionConsumerServiceURL attribute in Authn-Request sent by SP. ECP check this value with the ResponseConsumerURL in the paos:Request header in the authentication request message sent by SP. If the values are not the same, there is a possibility of man-in-the-middle attack that is some unauthorized external entity behaves like SP to obtain the assertions about the subject.

4.2.2.4 Processing Response

ECP forwards the SAML Response element to SP with PAOS binding. The Figure 4.12 shows the SOAP message that ECP sends to SP. paos:Response header is required according to the ECP Profile. If messageID attribute is used in paos:Request header in authentication request message (as we use in our example), refToMessageID attribute is also required in paos:Response header for message correlation.

The first thing that SP must do is verifying the signatures present on the SAML Assertions and Response elements. SP should also authenticate the IDP that is it should check if IDP is in the trust list of SP. However this is not

mentioned explicitly in the ECP profile.

In the implementation, the IDP is identified from the Issuer element in the Assertions (which is a required element according to ECP Profile). As a requirement of the ECP Profile, there should be a trust relationship between the IDP and SP. Therefore as discussed in the SAML AuthnRequest section, like IDP, SP has also a list of trusted IDPs (IDPs in the affinity domain as in our model) with their IDs (Provider ID used in the SAML messages), endpoint addresses (IDP service endpoint) and certificates. To authenticate IDP, the system get the identified IDP's certificate and check if it is equal to the certificate in the signature.

SP should check the SubjectConfirmation element to confirm that the assertion is related to the given subject. However, regardless of the subject confirmation method, SP must check some attributes: the Recipient attribute should be equal to the SP's own assertion consumer URL and InResponseTo attribute should be equal to ID of the AuthnRequest. SP then should check the optional elements inside the Conditions element.

After all of these evaluations, the AuthnStatement element can be used to establish a security context with the user (given in the Subject). However, the SessionOnOrNotAfter attribute must be considered for the life-time of this established security context.

4.2.2.5 Discussions on SAML SSO Profiles

SAML ECP Profile is more suitable for IHE XUA rather than SAML Web SSO Profile since the latter is totally browser based. On the other hand, ECP Profile does not mention pre-generated assertions which are one of the given types of the assertions in IHE XUA profile. If an X-Service User wants to use pre-generated assertions (which are called unsolicited responses in SAML profiles), the requirements of the SP about the assertion must be known. One way to achieve this is using the SAML Metadata specification for the SPs and IDPs. SAML Metadata is briefly described in the following sections. Another way is

putting strong restrictions over the assertion content. In any case, there is a need for a profile (not browser based, can be a modification of ECP) for pre-generated assertions.

Another issue is the communication between the IDP and ECP as discussed in Section 4.2.2.2. If IDPs are considered as separate entities in the affinity domain, the interface (how to communicate the authentication information, how to initiate fresh authentications) should be defined by profiles.

IHE services may also need authorization mechanisms on the requested resources or services. In order to conclude such access control decisions, the authorization mechanisms need some attributes about the user (user role, email, etc). SAML provides Attribute Profiles for this purpose. In our scenario, we obtain such attributes by using the SAML Attribute profiles after obtaining the authentication information by using the ECP. Nonetheless, IHE may need a combined profile in which attribute values can be gathered from IDPs during the authentication.

ECP Profile facilitates this by using SAML Metadata. In the authentication request message, SP gives an identifier with AttributeConsumingServiceIndex attribute. This value is used by IDP to access the SAML Metadata document of SP (Publishing and resolution of Metadata documents are described in SAML Metadata [30] specification). In this Metadata document, SP defines the required attributes for its authorization service. IDP finds the values of these attributes and puts them in the SAML Response element as AttributeStatements. In this architecture, using SAML Metadata becomes a must for SPs. Therefore, some other way may be provided which can be the extension of AuthnRequest element in the way that it can include SAML Attribute elements to query for attribute values (that is SP can ask for attribute values).

Reporting the failures during the execution of the profile is also very important. SAML provides this information within the SAML Status element in its transactions. The StatusCode element gives the identifier for the status. These identifiers are defined in SAML Core specification and define some basic possible

statuses and failures in SAML transactions. However, the objective of the Status element is to report the statuses or failures to the requestor of the assertion (SP in ECP), not reporting them to users. Therefore failure alternatives in the selected profiles, the mapping of the SAML status code values to these failures and the way to report them to users should be identified.

4.2.3 Attribute Federation with SAML Attribute Profiles

SAML Attribute Profiles give the specifications about how to name attributes and how to compare them. SAML XACML Attribute Profile is used since the attribute values are used for access control decision in the scenario. LDAP Attribute Profile, UUID Attribute Profile and DCE/PAC Attribute Profile are other important attribute profiles in SAML. SAML Assertion Query/Request Profile gives the specification of requesting attributes from an Attribute Authority (AA).

The Figure 4.13 shows the SAML AttributeQuery element which SP sends to IDP in the body of a simple SOAP message. The Subject element gives the identifier of the subject that the attributes are requested for. Then each Attribute element states the name and name format of the requested attributes. As in the ECP profile, the communicating parties (SP as attribute requester and IDP as attribute authority in our case) must authenticate each other. Message correlation is handled by the ID and InResponseTo attributes as seen from the Figure 4.13 and the Figure 4.14.

The SAML Response for the attribute query is shown in the Figure 4.14. The AttributeStatement element inside the Assertion provides the attribute values for the requested attributes. In order to respond such attribute queries, the Attribute Authority (attribute authority is IDP in our scenario) must have the ability of resolving the attribute from the attribute name and providing the value for the resolved attribute. The Attribute Authority can use the SAML Metadata to provide the list of attributes it can handle. The attribute names must be common and unique for both requester and the Attribute Authority

```
<samlp:AttributeQuery xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
        ID="nARKciu2Hxlng53"
        IssueInstant="2006-10-03T06:31:06.778Z" Version="2.0">
        <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" SPProviderID="sp"/>
        <saml:Subject xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
                <saml:NameID>doe</saml:NameID>
        </saml:Subject>
        <saml:Attribute xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" FriendlyName="User
Role" Name="urn:oasis:names:tc:XACML:2.0:example:attribute:role"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
        <saml:Attribute xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" FriendlyName="User
Email" Name="urn:oasis:names:tc:XACML:2.0:example:attribute:email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
        <saml:Attribute xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" FriendlyName="User
Name" Name="urn:oasis:names:tc:XACML:2.0:example:attribute:subject-name"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                ...
        </ds:Signature>
</samlp:AttributeQuery>
```

Figure 4.13: SAML Attribute Query

sides for attribute resolution.

IHE IT Infrastructure Planning Roadmap 2004-2009 Beyond has mentioned future profile candidates for enterprise and cross enterprise RBAC systems. In addition, Basic Patient Privacy Consent (BPPC) in PCC framework profile defines a way of using more than one privacy policy in an affinity domain. SAML Attribute profiles may play a major role in these profiles. They can either be used independently or with combination of XUA as discussed in the section 2.1.5. However, there are some open issues which should be decided by these profiles while using the SAML Attribute Federation mechanism.

First of all, these profiles should determine;

- Required and optional attributes for the profile,
- From which entity (Attribute Authorities) these attributes can be gathered; trusted entities like IDP, self assertions, other services like Personal White Pages (PWP) directory,
- How these attributes can be registered to the attribute authorities,
- How the attributes can be named (selecting appropriate SAML Attribute

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" ID="_b2tgT18I7s71AO7"
InResponseTo="nARKciu2Hxlng53"
IssueInstant="2006-10-03T06:31:07.247Z" Version="2.0">
        <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" SPProviderID="idp"/>
        <samlp:Status>
                <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
        </samlp:Status>
        <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_87y4RjTPbwpFr0e"
IssueInstant="2006-10-03T06:31:07.247Z" Version="2.0">
                <saml:Issuer SPProviderID="idp"/>
                <saml:Subject>
                        <saml:NameID>doe</saml:NameID>
                </saml:Subject>
                <saml:AttributeStatement>
                        <saml:Attribute FriendlyName="User Role"
                                Name="urn:oasis:names:tc:XACML:2.0:example:attribute:role"
                                NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
                                <saml:AttributeValue
xmlns:xs="http://www.w3.org/2001/XMLSchema"    xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xs:String">DIETICIAN</saml:AttributeValue>
                        </saml:Attribute>
                        <!-- Other Attributes-->
                </saml:AttributeStatement>
                <ds:Signature>
                        ...
                </ds:Signature>
        </saml:Assertion>
</samlp:Response>
```

Figure 4.14: SAML Attribute Statement

Profile).

Some attributes need to be updated very often rather than other static attributes like demographic information for a healthcare professional. Functional role of a healthcare professional on a patient is such an attribute which can not be simply stored forever. The information about these attributes is located in the information system of the healthcare enterprise and this information should be opened to outside by some services. In such situations using self assertions seem suitable. Another problem is that the SAML AttributeQuery can not handle querying such dynamic attributes. While requesting the attributes, only name of the attribute is stated in the SAML Attribute element. However, the attribute can depend on a three sided relationship which is subject-attribute-object relationship rather that subject-attribute relationship. For the above case, we should somehow state the patient identifier (like we give the subject identifier) in AttributeQuery to request the functional role of the professional

on the patient.

4.2.4 Usage of SAML Metadata

SAML Metadata [30] specification defines a way for SAML entities to agree and share system identifiers, endpoints, supported profiles, certificates and keys. It also defines a way to publish and find these metadata definitions. In this section the metadata definitions for the IDP and the SP that can be used in the implementation scenario is illustrated.

The Figure 4.15 shows the metadata of the IDP. The root element is the EntityDescriptor with the entityID attribute stating the unique identifier of the entity in the domain in which all these SAML issues are performed. The ds:Signature element is just to protect the integrity of the metadata definition. The last element, Organization, presents the basic information about the entity. The other elements under the root element, the IDPSSODescriptor and the AttributeAuthorityDescriptor, present the roles that the entity can play in the SAML profiles and protocols.

The IDPSSODescriptor element states the details of the role of IDP in the SSO profile (ECP profile in our case). For example, the WantAuthentication-RequestsSigned attribute states that SP must sign the AuthnRequest elements which is left optional in the ECP profile. The IDP's certificate which is used to sign the Assertions in the IDP's response message is given with the KeyDescriptor element. The SP may use the information in this element to retrieve the IDP's certificate for signature verifications and IDP authentication. In the service elements, the SingleSignOnService in this example shows the details of the services provided by a specified role. The endpoint and binding of the service is given by the Location and Binding attributes of the element. As mentioned before, if it is desired to transfer attribute values during the ECP Profile, IDP may state the supported attributes within this element as it is presented in the AttributeAuthorityDescriptor.

The AttributeAuthorityDescriptor element describes the Attribute Authority

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   entityID="idp">
   <ds:Signature>...</ds:Signature>
   <IDPSSODescriptor WantAuthnRequestsSigned="true"
         protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
         <KeyDescriptor use="signing">
            <ds:KeyInfo><ds:X509Data><ds:X509Certificate> ...
</ds:X509Certificate></ds:X509Data></ds:KeyInfo>
         </KeyDescriptor>
         <SingleSignOnService
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
            Location="http://144.122.230.23:9091/identity-provider/IdentityProvider"/>
           <!--Attributes supported in the ECP Response -->
           <!--Same as the Attributes in AttributeAuthorityDescriptor-->
   </IDPSSODescriptor>
   <AttributeAuthorityDescriptor
         protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
         <KeyDescriptor use="signing">
            <ds:KeyInfo><ds:X509Data><ds:X509Certificate> ...
</ds:X509Certificate></ds:X509Data></ds:KeyInfo>
         </KeyDescriptor>
         <AttributeService
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
            Location="http://144.122.230.23:9091/identity-provider/AttributeAuthority"/>
         <saml:Attribute
            NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
            Name="urn:oasis:names:tc:XACML:2.0:example:attribute:role"
            FriendlyName="UserRole">
            <saml:AttributeValue>ADMINISTRATIVE STAFF</saml:AttributeValue>
            <saml:AttributeValue>DIETICIAN</saml:AttributeValue>
            <saml:AttributeValue>MEDICAL DOCTOR</saml:AttributeValue>
            <saml:AttributeValue>NURSING STAFF</saml:AttributeValue>
            <saml:AttributeValue>PHARMACIST</saml:AttributeValue>
            <saml:AttributeValue>RESEARCHER</saml:AttributeValue>
         </saml:Attribute>
         <saml:Attribute
            NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
            Name="urn:oasis:names:tc:XACML:2.0:example:attribute:email"
            FriendlyName="UserEmail"/>
         <saml:Attribute
            NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
            Name="urn:oasis:names:tc:XACML:2.0:example:attribute:subject-name"
            FriendlyName="UserName"/>
   </AttributeAuthorityDescriptor>
   <Organization>
         <OrganizationName xml:lang="en">METU Identity Provider </OrganizationName>
         <OrganizationDisplayName xml:lang="en">
            Ankara - IHE Affinity Domain - METU Identity Provider
         </OrganizationDisplayName>
         <OrganizationURL xml:lang="en">http://www.srdc.metu.edu.tr</OrganizationURL>
   </Organization>
</EntityDescriptor>
```

Figure 4.15: Metadata of Identity Provider

service which is also used in the scenario. The Attribute elements inside the AttributeService define the supported attributes by IDP with their names and possible values.

The metadata of the SP is illustrated in the Figure 4.16. The AssertionCon-

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  entityID="https://ServiceProvider.com/SAML">
  <ds:Signature>...</ds:Signature>
  <SPSSODescriptor AuthnRequestsSigned="true"
        protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
        <KeyDescriptor use="signing">
          <ds:KeyInfo>
                <ds:X509Data><ds:X509Certificate> ... </ds:X509Certificate></ds:X509Data>
          </ds:KeyInfo>
        </KeyDescriptor>
        <AssertionConsumerService index="0"
          Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"
          Location="https://144.122.230.23:8443/xdsServices/xdsRep/AssertionConsumer"/>
        <AttributeConsumingService index="0">
          <ServiceName xml:lang="en">IHE XDS Retrieve Document</ServiceName>
          <RequestedAttribute
                NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                Name="urn:oasis:names:tc:XACML:2.0:example:attribute:role"
                FriendlyName="UserRole">
                <saml:AttributeValue>ADMINISTRATIVESTAFF</saml:AttributeValue>
                <saml:AttributeValue>DIETICIAN</saml:AttributeValue>
                <saml:AttributeValue>MEDICAL DOCTOR</saml:AttributeValue>
                <saml:AttributeValue>NURSING STAFF</saml:AttributeValue>
            <saml:AttributeValue>PHARMACIST</saml:AttributeValue>
            <saml:AttributeValue>RESEARCHER</saml:AttributeValue>
          </RequestedAttribute>
          <RequestedAttribute
                NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                Name="urn:oasis:names:tc:XACML:2.0:example:attribute:email"
                FriendlyName="UserEmail"/>
          <RequestedAttribute
                NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                Name="urn:oasis:names:tc:XACML:2.0:example:attribute:subject-name"
                FriendlyName="UserName"/>
        </AttributeConsumingService>
  </SPSSODescriptor>
  <Organization>
        <OrganizationName xml:lang="en">METU XDS Repository</OrganizationName>
        <OrganizationDisplayName xml:lang="en">
          Ankara IHE Affinity Domain - METU XDS Repository
        </OrganizationDisplayName>
        <OrganizationURL xml:lang="en">http://www.srdc.metu.edu.tr</OrganizationURL>
  </Organization>
</EntityDescriptor>
```

Figure 4.16: Metadata of Service Provider

sumerService gives the endpoint and binding of the assertion consumer at the SP side. Furthermore, the AttributeConsumingService element defines the required attribute values for the XDS Retrieve transaction. These attributes are used for auditing and authorization services in our implementation. As discussed before, these attributes can be either included in the IDP's response in ECP Profile or requested with the Assertion Query/Request Profile (as it is done in our implementation). The AttributeValue elements within the UserRole attribute states

the possible values for the user role attribute.

4.3 Access Control Model for XDS

In this section we continue giving examples from the implementation. How Extensible Access Control Markup Language (XACML) [13] and its RBAC profile [37] can be used to express consent policy and to implement the authorization service for the XDS Retrieve Document transaction is described. In addition, IHE Basic Patient Privacy Consent (BPPC) [20] profile and how XACML can be used within the profile is discussed.

4.3.1 XACML Model

XACML is an XML based mark-up language for the policy management and access decisions. XACML standard not only gives the model of the policy language, but also proposes a processing environment model to manage the policies and conclude the access decisions. In addition, it defines the Request/Response protocols for the communication between the application environment and the policy decision environment.

XACML represents the access control rules based on four main structures: Subject, Resource, Action, and Environment. Basic data source for defining the policy and the Request and Response messages are the attributes related with these main structures. For example, variety of the Subject attributes like name of the subject, email of the subject, role of the subject (both functional and structural), etc can be used in policy decisions. The Resource attributes may be resource ID or classification of the resource in terms of some criteria like sensitivity or type of the document. In addition, other attributes may be used like owner of the document (e.g. patient), time it is created or submitted to the system, the author institution or the author itself if we consider the Healthcare domain. The Action attributes are generally single identifiers like 'read', 'write', 'update' or 'delete'. However, the action can also be defined with more than one attributes in more complex situations. The Environment attributes defines

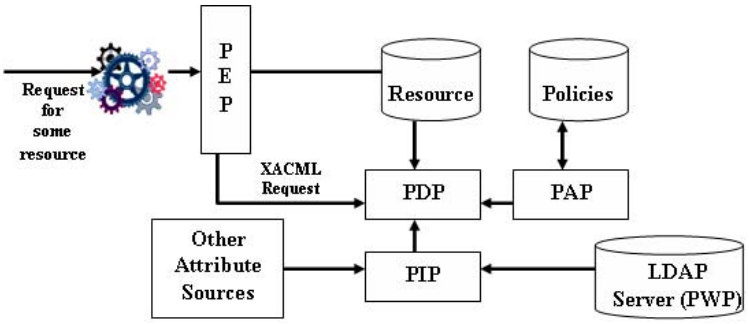

Figure 4.17: XACML Processing Environment

the current environment which is independent of other structures (e.g. current time).

Another source of data to use in XACML policy decisions and transactions is the requested resource itself. XACML use XPath standard for the XML based resources. For example, an EHR document may include demographic data of the patient and the access policy may have a statement about the age of the patient. In this case, the age information may be retrieved by XPath expressions given in the policy definitions. Furthermore, in this way, not the whole resource but the parts which are permitted by the XACML policy can be provided to the requester. However, in order to perform such operations, the structure of the resource must be fixed, it must be XML and it must be known by the authors of the XACML policy. For the healthcare domain, this can be achieved only for strictly regulated and specialized systems in terms of content and communication.

The XACML specification also defines a processing environment model as illustrated in the Figure 4.17. Four actors are mentioned in this processing environment; Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Administration Point (PAP) and Policy Information Point (PIP).

PEP is the entry point for the access control mechanism which isolates the XACML processing environment from the application environment. It requests

the access decision from the PDP by sending an XACML Request message. The Request message includes the attributes about the resource, subject and action as described above. These attribute values are obtained from the service which requests the resource. SAML technology and assertions are the best example for cross enterprise services which shows how such information can be obtained. However, providing all attributes in the Request message is not dynamic since the PEP should know all the required attributes for the execution of policy before asking for the decision. XACML specification proposes the PIP entity for this purpose. When the PDP needs value of an attribute it asks the PIP which knows how to obtain the values from outside services or from attribute values exist in the XACML Request. In our implementation, we simply assume that the required attributes for the consent policies are invariable and known by the PEP. Therefore, the PEP obtains the attribute information by using the SAML Assertion Query/Request profile from the IDP and provides the attribute values within the XACML Request. As seen from the Figure 4.17 and above discussions, several possible alternative access control architectures can be provided by using the XACML and SAML (most suitable for XACML) standards. In the following sections more discussions about how XACML can be used for IHE BPPC and further consent or privacy policy related profiles are discussed.

4.3.1.1 Sample Access Control Policies for XDS

Today access control models are mostly based on RBAC model. XACML also has published an RBAC profile which defines how the XACML can be used to construct RBAC policies. This profile divides the policies into two types; Role policies and Permission policies. The Figure 4.18 and the Figure 4.19 shows the example Role and Permission policies for the XACML RBAC profile.

The Figure 4.18 illustrates a Permission PolicySet which defines the rules to access the medical records which are annotated by the 'GeneralClinicalInformation' value as their sensitivity level. As seen from the example, the sensitivity

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os" PolicySetId="PPS:MEDICALDOCTOR:Role"
PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
 <Target/>
 <Policy PolicyId="Policy1" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-
overrides">
  <PolicyDefaults><XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion></PolicyDefaults>
  <Target>
   <Resources>
        <Resource>
         <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">GENERALCLINICALINFORMATION</AttributeValue>
          <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:confCode"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </ResourceMatch>
        </Resource>
   </Resources>
  </Target>
  <Rule RuleId="Role:ConfCode:Rule1" Effect="Permit">
   <Target/>
   <Condition>
        <Apply FunctionId="urn:oasis:names:tc:xacml:2.0:function:time-in-range">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
          <EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"
DataType="http://www.w3.org/2001/XMLSchema#time"/>
         </Apply>
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">08:00:00+02:00</AttributeValue>
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">20:00:00+02:00</AttributeValue>
        </Apply>
   </Condition>
  </Rule>
  <Obligations>
   <Obligation ObligationId="tr:edu:metu:srdc:xds:pep:obligations:mail" FulfillOn="Permit">
        <AttributeAssignment AttributeId="tr:edu:metu:srdc:xds:attribute:mailto"
DataType="http://www.w3.org/2001/XMLSchema#string">
         <ResourceAttributeDesignator AttributeId="tr:edu:metu:srdc:xds:attribute:patient-email"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </AttributeAssignment>
        <!--Other required attributes for mail obligation-->
   </Obligation>
  </Obligations>
 </Policy>
</PolicySet>
```

Figure 4.18: Permission PolicySet

value of the resource is obtained from the xacml:1.0:resource:confCode attribute which is a resource attribute in the XACML Request message coming to the PDP.

The XACML Condition element defines the rules to access the specified type of resource. In our example, it states that GeneralClinicalInformation are accessible only between hours 08:00 and 20:00. The Obligation element states that if the decision is permited for this Permission Policy then the obligation with the identifier tr:edu:metu:srdc:xds:pep:obligations:mail must be realized. The Obligation concept provides some functionality to the system and policy makers to define responsibilities and obligations for the system or the requester of the resource which must be obeyed after the decision is taken. For example,

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
PolicySetId="RPS:MEDICALDOCTOR:Role" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
combining-algorithm:permit-overrides">
 <Target>
  <Subjects>
   <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
         <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MEDICALDOCTOR</AttributeValue>
         <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:role"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </SubjectMatch>
   </Subject>
  </Subjects>
 </Target>
 <PolicySetIdReference>PPS:MEDICALDOCTOR:Role</PolicySetIdReference>
</PolicySet>
```

Figure 4.19: Role PolicySet

the obligation that is shown in the Figure 4.18 forces the system to inform the patient with an email after the resource access is granted. The attributes that are used to perform the obligation are also included in the obligation (e.g. the mail address of the patient). Obligations are processed by the PEP.

The Role PolicySet is illustrated in the Figure 4.19. It requires matching the urn:oasis:names:tc:xacml:1.0:subject:role attribute to the value 'MedicalDoctor' for the applicability of the PolicySet. The PolicySetIdReference elements give references to the Permission PolicySets which are processed further to decide on the access decision for the subject with the 'MedicalDoctor' role.

4.3.2 BPPC Discussions

Our scenario has some differences with the IHE BPPC Profile in terms of consent management model. In our scenario as described before, patients create their privacy consent policies which are then sent to XDS Repository. Then, when a document is requested, the privacy consent policy which the patient gives to the institute is found by the XDS Repository and used for the access control decision. This scenario is not practical in real life in terms of legal issues (An institute may not be able to get consent for the future documents. As another case, a consent policy is needed to be created by the patient for a specific document or a

group of document which may be a tedious case). However, it may be beneficial in some special cases where patient has to define some access control rules for specific health records as mentioned in one of the BPPC use case; 'Policies in an environment with comprehensive access controls'. On the other hand, in the normal use cases of the BPPC profile, the Privacy Consent Policies are provided by the policy makers of the Affinity Domain. The patients select some of these policies and sign a consent document that references to these consented policies. The healthcare institutes in the affinity domain must obey the rules in the consented policies.

The enforcement point for the access decision is located at the client side (Document Consumers) in BPPC profile. In our implementation the access control enforcement is performed at the service side (XDS Repository). In this respect, the BPPC profile assumes a strong trust on the document consumer systems for applying the Privacy Consent Policies. With this assumption, the system becomes more simple since the information (attributes, user identification, etc) for access control mechanisms that is located at the client side does not need to be transferred to the service side. Nevertheless, this type of information is needed for auditing. In fact, this assumption can not be accepted by some other business domains in terms of security and privacy requirements. However, when we consider the situation together with the IHE affinity domain concept and since the enforcement is related with patient consents, the trust assumption seems reasonable and practicable. On the other hand, the Service Providers (XDS Repositories, XDS Registries, PIX Managers, etc) in the affinity domain should have their own privacy policies and access control mechanisms which may require the same user information from the client side.

BPPC does not restrict the content of the Privacy Consent Policies. Therefore, the implementations of the access control mechanisms are manual and specific to the Privacy Consents defined in the Affinity Domain. On the other hand, the implementation of the mechanisms will be very easy if IHE selects a machine processable access policy standard for the Privacy Consent Policies.

The XACML standard seems to be very suitable for this purpose. It can provide all functionalities for the access control systems mentioned in the BPPC profile. In addition, it also supports more complex functionalities for future refinements and the profiles. The following section discusses the use of the XACML to represent the Privacy Consent Policies of the affinity domain mentioned in the BPPC profile.

4.3.2.1 Using XACML for Privacy Consent Policies

The BPPC Profile provides a possible implementation way of Privacy Consent Policies. It uses an access control matrix consisting of roles and sensitivity markers. The matrix can be sliced in several ways to form the Privacy Consent Policies. The XACML language can be used in any way that is used to divide the access control matrix.

The first example is using policies which describe the whole access control matrix. In this way several access control matrices are generated and each of them is represented by single Privacy Consent Policy describing the preferences about all the sensitivity markers. The patient should select only one of them since the policies describe different matrixes and they may be incompatible. Using such a methodology may not be preferred since the description of the policy is difficult and complex to make the patients understand them. If such a methodology is used, the Privacy Consent Policies should be named or classified in terms of their restrictive capabilities (e.g. loose, restrictive, very restrictive, etc).

If the matrix is divided based on either the role vocabulary or sensitivity markers, the methodology defined in the XACML RBAC profile can be used. In this case, the Privacy Consent Policies put restrictions for a role defining which sensitivity markers the role is allowed to access or for a sensitivity marker defining which roles are allowed to access the resource for the defined sensitivity marker. The example for the sensitivity marker based slicing is given since naming the Privacy Consent Policies with the sensitivity markers is more suitable

```
<PolicySet PolicySetId="urn:ihe:bppc:privacyconsentpolicies:example:SensitiveInformation"
PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:only-one-applicable ">
 <Target>
  <Resources>
   <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
         <AttributeValue>SensitiveInformation</AttributeValue>
         <ResourceAttributeDesignator AttributeId="..."/>
        </ResourceMatch>
   </Resource>
  </Resources>
 </Target>
 <!-- Obligations for the Privacy Consent policy in case of the decision is permit or deny-->
 <PolicyIdReference>
    urn:ihe:bppc:rolepolicies:example:DirectCareProvider
 </PolicyIdReference>
 <PolicyIdReference>
   urn:ihe:bppc:rolepolicies:example:EmergencyCareProvider
 </PolicyIdReference>
</PolicySet>
```

Figure 4.20: Privacy Consent Policy defined by XACML

(it is more meaningful to publish resources with the confidentiality code corresponding to sensitivity markers). As it is shown in the examples, the policies are divided into two types. However, references are given from Permission policies to Role policies that is Permission Policies are executed first. Other restrictions (e.g. time) can be put on the Role policy.

The Figure 4.20 illustrates a Privacy Consent Policy for an Affinity Domain defined by XACML. The PolicySetId can be used as a unique Privacy Consent Policy identifier. The XACML Target element gives the information in order to decide if the PolicySets or Policies are applicable or not. In our example, if the resource is not classified as 'SensitiveInformation' then this Privacy Consent Policy is not applicable. If the Privacy Consent Policy is applicable, the role policies given by references will be executed. The policy combining algorithm defines the way of execution and combination of the results. The Only-One-Applicable algorithm states that only one policy can be selected as applicable and overall result is the result of the applicable policy. The Figure 4.21 shows the Role policy which is referenced from the Privacy Consent Policy. If the target matches then the rules (other restrictions) will be evaluated and result is returned. Any Rule combining algorithm can be selected according to the

```
<Policy PolicyId="urn:ihe:bppc:rolepolicies:example:DirectCareProvider"
RuleCombiningAlgId="Depends-On-Your-Choice-Over-Rules">
 <Target>
  <Subjects>
   <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
         <AttributeValue>DirectCareProvider</AttributeValue>
         <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:role"/>
        </SubjectMatch>
   </Subject>
  </Subjects>
 </Target>
 <!-- Rules: other restrictions-->
 <!-- Obligations-->
</Policy>
```

Figure 4.21: Role Policy for BPPC

rules and preferences defined in the policy. If there are no restrictions, a single rule stating the rule effect as Permit is enough. The corresponding obligations in Role Policies and the general obligations defined in Privacy Consent Policy should be executed after taking the decision.

Other methodologies can also be produced by defining different policy combining algorithms. Currently, XACML has provided six policy or rule combining algorithms. The main algorithms are Permit-Overrides, Deny-Overrides, First-Applicable, Only-One-Applicable. The XACML RBAC uses permit overrides to combine the permission policies. In the above example, we use Only-One-Applicable algorithm since we assume a user can have only one role. The discussion can also be applied to Privacy Consent Policies. If it is assumed to have a document with more than one sensitivity marker, then an appropriate policy combining algorithm must be chosen to combine the Privacy Consent Policies. However, BPPC profile does not allow multiple roles or multiple sensitivity markers.

The Document Consumer actors which implements BPPC profile with such Privacy Consent Policies in XACML format should execute the following steps:

- Find the Privacy Consent Policies (XACML PolicySets) which the patient has given consent.

- Combine them to single PolicySet with Only-one-applicable algorithm

- Determine the access grant;

  - a-If the PolicySet evaluates to Permit, grant access
  - b-If the PolicySet evaluates to NotApplicable, deny access
  - c-If the PolicySet evaluates to Deny, deny access

The XACML evaluation models are based on matching of the conditions in the rules and targets in policy and policy sets. The policy maker can only state 'deny' or 'permit' decision while giving the effect of a rule. For example in the above example, if an administrator tries to access a record classified as 'SensitiveInformation', then the target in the PolicySet shown in Figure 4.20 matches. However, the two role policies do not match with the 'Administrator' role. In this case, the PolicySet evaluates to 'NotApplicable', not to 'Deny'. To produce the 'Deny' result, the PolicySet should include the 'Administrator' role policy and this policy has a rule with effect value 'Deny'. Such rules are called negative-rules. However, negative rules are not recommended by authorities since they can lead to policy violations. The XACML support negative rules but it recommends not to use them. The BPPC profile also mentions negative rules and states that they must not be used. Therefore, the step 3.c never occurs if negative rules are not used and the 'Not Applicable' result (3.b) implicitly defines a deny situation.

4.4 Auditing Healthcare Events

Some details about IHE ATNA profile is already given in Chapter 2 and Chapter 3. This section illustrates the component, Audit Record Repository Server, which I have implemented in the scope of this thesis work.

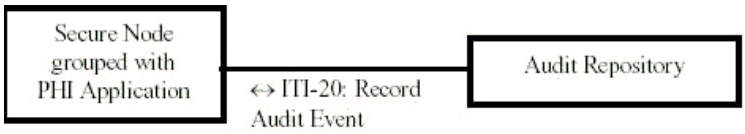

Figure 4.22: ATNA Record Audit Event Transaction

4.4.1 Audit Record Repository Server

The Audit Record Repository Server component is a fully conformant implementation for the Audit Record Repository actor of IHE ATNA Profile. It serves as an interface to clients to which they can send the audit records they produced in their systems. The Figure 4.22 illustrated the high level transaction that IHE defines to enable clients to record their audit events.

As already mentioned, the IHE Audit Record Repository Actor should support two communication protocols to carry the audit message.

- *Reliable Syslog Cooked Profile (RFC-3195):* This profile is based on the BEEP profile [38] which is a generic application protocol framework for connection-oriented, asynchronous, interactions. Within BEEP, features such as authentication, privacy, and reliability through retransmission are provided. The Cooked Profile provides a structured entry format for the audit which also provide acknowledgement of both sides.

- *BDS Syslog (RFC-3164):* The BSD Syslog protocol has some disadvantages according to RFC-3195. It does not provide acknowledgement, authentication of the sender and encryption of the audit message. In addition, the messages may be lost or truncated.

Reliable Syslog Cooked Profile is an extension to BSD Syslog and they share some basic properties like the syslog header part. The Figure 4.23 shows a sample syslog header which is called Syslog Entry element in RFC3195. The syslog header is composed of five attributes; facility, severity, timestamp, host

Figure 4.23: RFC3164 Syslog Header

and tag.

The facility and severity attributes are coded in the PRI part which starts with < and ends with >. The Facility value gives the type of the message that is which system it is originated. Some facility codes defined in the specification are kernel messages, user-level messages, and mail system. The Severity code gives the severity of the message like Emergency:system is unusable, Alert:action must be taken immediately, and Debug level messages.

After the PRI part a timestamp is required which shows the origination time of the message. The host part, given as kadikoy.srdc.metu.edu.tr in the figure, is optional in the header. It shows the IP or host name of the system that generates the audit message. The tag part gives an identifier for the system which produces the message. For example, the XDS_SRDC value given in the Figure is the identifier of the XDS Registry/Repository System deployed in SRDC.

The IHE ATNA profile use the Security Audit and Access Accounability Message XML Data Definitions for Healthcare Applications (RFC-3881) [23] which defines the content of the Audit Record with an XML schema. The ATNA profile states that only the events which can be defined by RFC-3881 should be reported. ATNA profile also defines the events that should be reported to Audit Record Repository.

The Figure 4.24 shows a sample audit record which shows the details of event Audit Log Used which is a required event that should be reported by Audit Record Repository when a person views the audit record. An RFC3881 compliant audit record is composed of four sub elements which we describe

below.

- *Event Identification:* It gives information about the event that the audit message is generated for. The EventID subelement provides the identifier of the event. For instance, in our sample it is Audit Log Used which is coded according to code system given int the DICOM Supplement 95 (DCM). EventDateTime gives the time of the event. The outcome indicator provides the information whether event is successful or not.

- *Active Participant:* It defines an actor which contributes to the event. This actor can be an human (a user) or a system (software) which takes part in the event. In our example, the actor is a user who uses the audit log. As shown in figure, ID and role of the user is given by codes.

- *AuditSourceIdentification:* This element gives the identifier of the system who sends the audit message.

- *ParticipantObjectIdentification:* It gives information about the objects which are affected from the event. For instance, a patient or a medical record can be a ParticipantObject. In our example the audit log with the identifier 38 is the object of the event.

The RFC-3881 provides also some vocabulary which can be used in the audit record. In addition, it defines the trigger events that can occur in the healthcare systems. The following list the vocabularies that RFC-3881 defines.

- *EventActionCode:* Action in the event (e.g. read, update, delete)

- *EventOutcomeIndicator:* Result of the event (e.g. success, minor failure, major failure)

- *AuditSourceTypeCode:* Defines the type of the Audit Producing System (e.g. end-user display device, web server process, application server process)

```
<AuditMessage>
    <EventIdentification EventOutcomeIndicator='0' EventDateTime='2007-05-09T10:20:30'
EventActionCode='R'>
        <EventID displayName='Audit Log Used' code='110101' codeSystemName='DCM'/>
    </EventIdentification>
    <ActiveParticipant NetworkAccessPointTypeCode='2' UserID='tuncay'
NetworkAccessPointID='144.122.230.23' UserIsRequestor='true'>
        <RoleIDCode displayName='ARR_SRDC-Users' code='0' codeSystemName='METU' />
    </ActiveParticipant>
    <AuditSourceIdentification AuditSourceID='ARR_SRDC' />
    <ParticipantObjectIdentification ParticipantObjectTypeCode='2' ParticipantObjectDataLifeCycle='6'
ParticipantObjectTypeCodeRole='13' ParticipantObjectID='38'>
        <ParticipantObjectIDTypeCode displayName='Report Number' code='9'
codeSystemName='RFC3881' />
        <ParticipantObjectName>AuditLog_38</ParticipantObjectName>
    </ParticipantObjectIdentification>
</AuditMessage>
```

Figure 4.24: Sample RFC3881 Audit Log

- *NetwrokAccessPointTypeCode:* The type of end point address for the service (e.g. machine name, IP address, telephone number)
- *ParticipantObjectIDTypeCode:* Defines the type of identifier of the object which participates to event. (e.g. patient number, encounter number, report number)
- *ParticipantObjectTypeCode:* Defines the type of the participated object. (e.g. person, system object, organization)
- *ParticipantObjectTypeCodeRole:* Defines the role of the participated object. (e.g. patient, user, doctor, resource)
- *ParticipantObjectDataLifeCycle:* Defines the life-cycle for the participated object. (e.g. origination/creation, import, translation, verification)

Message header and message content is parsed and printed separately in the command line interface of the Audit Record Repository. A graphical user interface is also developed in Software Research and Development Center for this Audit Record Repository.

CHAPTER 5

CONCLUSION

Privacy is a matter of individual liberty, autonomy, and even a fundamental human right. All these perspectives are strongly applicable in health context. On the other hand, breaches of confidentiality are harmful because they can lead to privacy protective behavior, in which patients avoid seeking health care in order to protect their personal information. Such behavior has devastating effects on both individual health and, more generally, on public health. This is just one important reason why we need to build confidentiality and security into a networked environment.

There are several initiatives in the EU Member States to establish networked electronic health information environment for sharing EHRs. These efforts are global, for example the National Healthcare Information Network (NHIN) initiative of USA and the Health Infoway initiative in Canada have invested huge amounts of money in order to establish such networks. The emergence of a networked electronic health information environment will transform patient care and improve the efficiency and effectiveness of the health system. At the same time, the emerging electronic health information infrastructure and the massive increase in the volume of health data that is easily collected, linked, and disseminated create unprecedented privacy and security risks that needs to be adequately and appropriately addressed. Although some of these risks exist in

an offline world, they have become more pronounced for networked environments due to the scale of data transactions and the relatively greater ease of collecting, linking, and disseminating information over the networks.

IHE XDS profile is becomeing very popular for emerging health networks. Both NHIN and Canada Health Infoway initiatives have selected the IHE XDS as the basis of their health information network. Therefore, privacy and security issues in IHE XDS affinity domain becomes important research and development area for health IT vendors and professionals. This work proposes a basic infrastructure by using current standards. The health IT players and IHE technical committees and standard bodies can benefit from the implementation experiences and recommendations provided in this work, to propose future interoperability profiles or design systems related with the mentioned issues. The following items summarize the basic issues discussed in this work regarding the usage of standards like SAML and XACML:

- In the Identity Federation architectures (both in the Web Service efforts and SAML specifications), the trusted intermediaries are the basic actors to federate the identity and trust among the service providers and the service clients. Therefore, the XUA profile should specify a model describing the relationships between service providers, service clients and the trusted intermediaries (Identity Providers in SAML specifications and Security Token Service in WS-Trust model) in an affinity domain. Furthermore, it should specify how this model can be extended for the federation of affinity domains.

- In the ECP Profile, the Identity Provider (IDP) should be able to identify the subject (principal) in order to give an assertion about the subject. Therefore, either there should be prior established security context between the Identity Provider (IDP) and the Enhanced Client (ECP) or the context should be established by initiating a fresh authentication. However, the ECP Profile does not specify any method or communication flow for this purpose. In our implementation we use cookies. In this respect,

in order to provide interoperability, XUA profile should specify standard methodologies.

- In the XUA profile, the content of the SAML assertions are not specified clearly. In the ECP profile, SAML assertions can carry authentication and attribute statements. However, the requested attributes cannot be specified in SAML AuthnRequest message which is sent by the Service Provider. The ECP Profile can be extended in this respect. Furthermore, the SAML attribute query mechanisms do not handle more complex attribute queries (e.g. asking a functional role of a professional for a patient).

- Some of the procedures, elements and attributes are left optional in SAML Profiles. By using the SAML Metadata specification, actors using the SAML framework can define their choice for the optional items. In addition, these actors can define their requirements (e.g. required attributes for authorization) and preferences regarding the SAML profiles. The SAML Metadata specification can be recommended for the actors implementing the XUA profile to define the related metadata needed in SAML framework.

- After getting the SAML Assertion, the Service Provider can use the authentication statement in the assertion to establish a security context with the subject. If such a context is not established, the same process should be repeated for each request for the service (e.g. a user may perform several XDS queries). SAML specifications do not specify a mechanism for this purpose. On the other hand, WS-SecureConversation can be used for web service transactions to establish security context in the Service Provider side.

- In the XUA profile while profiling the cross-enterprise user authentication, the WS-Trust specification will be the main element for web service transactions. The WS-Trust specification defines the basic building blocks (like SAML Core specification does) to construct a trust model. However, it

needs further profiles (like SAML Profiles) which define the usage of these building blocks to handle specific use-cases (e.g. single sign on).

- IHE BPPC profile does not restrict the content of the Privacy Consent Policies. Therefore, the implementations of the access control mechanisms will be manual and specific to the Privacy Consents defined in the Affinity Domain. On the other hand, the implementation of the mechanisms will be very easy if IHE selects a machine processable access policy standard for the Privacy Consent Policies. The XACML standard seems to be very suitable for this purpose. It can provide all functionalities for the access control systems mentioned in the BPPC profile. In addition, it also supports more complex functionalities for future refinements and the profiles.

REFERENCES

[1] HITSP EHR Interoperability Specification, http://www.ansi.org/hitsp/, Last accessed date, May 2007.

[2] Canada
Health Infoway, EHRS Blueprint, http://knowledge.infoway-inforoute.ca/-en/knowledge-centre/ehrs-blueprintv2.aspx, Last accessed date, January 2007.

[3] Integrated Healthcare Enterprises, http://www.ihe.net, Last accessed date, May 2007.

[4] IHE. Cross Enterprise Document Sharing Integration Profile, IHE IT Technical Framework Volume 1, http://www.ihe.net/Technical_Framework/-upload/ihe_iti_tf_3[1].0_vol1_FT_2006-11-07_tracked.pdf, November 2006.

[5] Health@net Project, http://www.healthatnet.at/index.php, Last accessed date, April 2007.

[6] Liberty Alliance Project, http://www.projectliberty.org/, Last accessed date, May 2007.

[7] OASIS. Security Assertion Markup Language, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security, Last accessed date, May 2007.

[8] OASIS. WS Security Core Specification v1.1, http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf, February 2006.

[9] OASIS. WS-Trust v1.3 , http://docs.oasis-open.org/ws-sx/ws-trust/-200512/ws-trust-1.3-rddl.html, Last accessed date, March 2007.

[10] OASIS. WS-SecureConversation v1.3, http://docs.oasis-open.org/ws-sx/-ws-secureconversation/200512/ws-secureconversation-1.3-rddl.html, Last accessed date, March 2007.

[11] Web Service Federation Language, ftp://www6.software.ibm.com/-software/developer/library/ws-fed.pdf, December 2006.

[12] D.R. Ferraiolo, D.F.; Kuhn. Role based access control. *15th National Computer Security Conference*, 1992.

[13] OASIS. Extensible Access Contrl Markup Language, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml, Last accessed date, May 2007.

[14] IBM. The Enterprise Privacy Authorization Language, http://www.zurich.ibm.com/-security/enterprise-privacy/epal/, Last accessed date, January 2007.

[15] The Internet Engineering Task Force, http://www.ietf.org/, Last accessed date, May 2007.

[16] Usman; Bertino Elisa; Ghafoor Arif Joshi, James B. D.; Latif. A generalized temporal role-based access control model. *IEEE Transactions on Knowledge and Data Engineering archive*, 17, 2005.

[17] J.M.; Moody K. Hayton, R.J.; Bacon. Access control in an open distributed environment. *Security and Privacy, Proceedings. IEEE Symposium*, 1998.

[18] R. El; Balbiani P.; Benferhat S.; Cuppens F.; Deswarte Y.; Mige A.; Saurel C.; Trouessin G. Kalam, A. Abou El; Baida. Organization based access control. *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, 2003. http://www.orbac.org/publi/OrBAC/OrBac.pdf.

[19] European Committee for Standardization. CEN EN 13606-4 Health informatics - Electronic health record communication - Part 4: Security, March 2007.

[20] IHE. Basic Patient Privacy Consent Integration Profile, http://www.ihe.net/Technical_Framework/upload/IHE_PCC_TF_BPPC-_Basic_Patient_Privacy_Consents_20060810.pdf, August 2006.

[21] The Open Group. Distributed Audit Service, http://ospkibook.sourceforge.net/docs/OSPKI-2.4.7/OSPKI-html/xdas.htm, Last accessed date, March 2007.

[22] IHE. Audit Trail and Node Authentication Integration Profile, http://www.ihe.net/Technical_Framework/upload/ihe_iti_tf_3[1].0_vol1_FT_2006-11-07_tracked.pdf, November 2006.

[23] Glenn Marshall. Security audit and access accountability message xml data definitions for healthcare applications, 2004. http://www.faqs.org/rfcs/-rfc3881.html.

[24] DICOM. Digital Imaging and Communications in Medicine Supplement 95, ftp://medical.nema.org/medical/dicom/supps/sup95_fz.pdf, June 2004.

[25] C.Lonvick. The bsd syslog protocol, 2001. http://www.faqs.org/rfcs/-rfc3164.html.

[26] Rose, M; New, D. Reliable delivery for syslog, 2001. http://www.faqs.org/-rfcs/rfc3195.html.

[27] IHE Cross Enterprise User Authentication (XUA) 2006-2007 White Paper, http://www.ihe.net/Technical_Framework/upload/-IHE_ITI_TF_White_Paper _CrossEnt_User_Authentication_PC_2006-08-30.pdf, June 2006.

[28] OASIS. Security Assertion Markup Language Bindings Specification, http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf, March 2005.

[29] OASIS. Security Assertion Markup Language Technical Overview, http://www.oasis-open.org/committees/download.php/20645/-sstc-saml-tech-overview-2-draft-10.pdf, October 2006.

[30] OASIS. Security Assertion Markup Language Metadata Specification, http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf, March 2005.

[31] OASIS. Security Assertion Markup Language Authentication Context Specification, http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf, March 2005.

[32] OASIS. Security Assertion Markup Language Core Specification, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, March 2005.

[33] SSL 3.0 Specification, http://wp.netscape.com/eng/ssl3/, Last accessed date, May 2007.

[34] TLS Protocol v1, http://www.ietf.org/rfc/rfc2246.txt, January 1999.

[35] OASIS. Security Assertion Markup Language Implementation Guidelines, http://www.oasis-open.org/committees/download.php/8958/sstc-saml-implementation-guidelines-draft-01.pdf, August 2004.

[36] OASIS. Security Assertion Markup Language Profiles, http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf, March 2005.

[37] OASIS. Extensible Access Control Markup Language Core and hierarchical role
based access control (RBAC) profile, http://docs.oasis-open.org/xacml/-2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf, February 2005.

[38] M. Rose. The blocks extensible exchange protocol core, 2001. http://www.rfc-editor.org/rfc/rfc3080.txt.