Privacy Trust Access Control Infrastructure Using XACML

Uche Magnus Mbanaso

Informatics Research Institute (IRIS)

School of Computing, Science and Engineering

University of Salford, Salford, UK

Submitted in Partial Fulfilment of the Requirements of the Degree of Doctor of Philosophy, April 2009

Acknowledgements

My sincere gratitude goes to my Supervisors: Professor David W. Chadwick and Professor Grahame Cooper for their excellent academic support and financial help during the course of my study.

I would like to thank Anne Anderson and Seth Proctor, Senior Staff Engineers of Sun Microsystems for their various contributions in the areas of XACML and SAML standards.

To the entire administrative staff of IRIS and the postgraduate team of the School of Computing, Science and Engineering, I would like to say big thanks for their various support.

I would also like to appreciate my friends Emmanuel Egwuatu, Sir Chris Ohadiugha and Dr Ken Omeje, and my Uncle Sir G.E Okafor for their encouragement. I thank Dr V. Ururuka and family for their prayers and support. Additionally, I thank Chief and Mrs Elias Ukachukwu for their various help.

To my family, particularly my wife Ada for proof-reading this thesis, my daughters Dili, Debby and Sally, and my Son Jesse Caleb, I give my profound thanks for their understanding, prayers and support and my senior brother Nicholas for his encouragement.

Finally, I owe my deepest gratitude to the Almighty God, by whose grace I have been able to carry out this work.

Declaration

The author acknowledges that the work described by this thesis involved the contributions of three other people besides him.

Professor Grahame Cooper provided useful guidance and remarks in designing the algorithms for the Obligation of Trust protocol message processes as well as in planning the structure of the thesis.

Professor David W. Chadwick provided useful advice and comments on the development of the concepts that underpinned the author's work.

Anne Anderson worked with the author to define the schema extensions of the SAML request/response protocol to support the Obligation of Trust protocol.

Abstract

The use of personal, sensitive information, such as privileges and attributes, to gain access to computer resources in distributed environments raises an interesting paradox. On one hand, in order to make the services and resources accessible to legitimate users, access control infrastructure requires valid and provable service clients' identities or attributes to make decisions. On the other hand, the service clients may not be prepared to disclose their identity information or attributes to a remote party without determining in advance whether the service provider can be trusted with such sensitive information. Moreover, when clients give out personal information, they still are unsure of the extent of propagation and use of the information. This thesis describes an investigation of privacy preserving options in access control infrastructures, and proposes a security model to support the management of those options, based on eXtensible Access Control Markup Language (XACML) and Security Access Markup Language (SAML), both of which are OASIS security standards. Existing access control systems are typically unilateral in that the enterprise service provider assigns the access rights and makes the access control decisions, and there is no negotiation between the client and the service provider. As access control management systems lean towards being user-centric or federated, unilateral approaches can no longer adequately preserve the client's privacy, particularly where communicating parties have no pre-existing trust relationship. As a result, a unified approach that significantly improves privacy and confidentiality protection in distributed environments was considered. This resulted in the development of XACML Trust Management Authorization Infrastructure (XTMAI) designed to handle privacy and confidentiality mutually and simultaneously using the concept of Obligation of Trust (OoT) protocol. The OoT enables two or more transaction parties to exchange Notice of Obligations (NoB) (obligating constraints) as well as Signed Acceptance of Obligation (SAO), a proof of acceptance, as security assurances before exchange of sensitive resources.

Table of Contents

Table of Figures

List of Important Terms and Abbreviations

Abbreviation	Definition
AA	Attribute Authority
ADF	Access Decision Function
AEF	Access Control Enforcement Function
AFM	Attribute Finder Module
ARP	Attribute Release Policy
ATN	Automatic Trust Negotiation
CA	Certificate Authority
CAS	Central Authentication Service
DES	Data Encryption Standard
DSA	Directory System Agent
DTD	Document Type Definition
EPAL	Enterprise Privacy Authorization Language
FIM	Federal Identity Management
FIP	Fair Information Practice
HTTP	Hypertext Transfer Protocol
IDEA	International Data Encryption Algorithm
IdP	Identity Provider
IPSec	Internet Protocol Security
LDAP	Lightweight Directory Access Protocol
NoB	Notification of Obligation
OASIS	Organization for the Advancement of Structured Information Standards
OECD	Organization for Economic Co-operation and Development

OoT	Obligation of Trust
P3P	Platform for Privacy Preference
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PII	Personal Identifiable Information
PIP	Policy Information Point
PKC	Public Key Certificate
PKI	Public Key Infrastructure
P_{UK}	Public Key
RAA	Role Activation Authority
RAD	Rapid Application Development
RBAC	Role-Based Access Control
RC2	Rivest Cipher 2
SAML	Security Assertion Markup Language
SAO	Signed Acceptance of Obligation
SDLC	Software Development Life Cycle
SOA	Source of Authority, or Service Oriented Architecture
SP	Service Provider
SSL	Secure Socket Layer
SSO	Single Sign On
STS	Security Token Service
TLS	Transport Layer Security
TN	Trust Negotiation
TTP	Trusted Third Party
UML	Unified Modelling Language
W3C	World Wide Web Consortium

WSA	Web Services Architecture
WSSE	Web Service Security Environment
WS-XACML	Web Services XACML
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language
XTMAI	XACML Trust Management Authorization Infrastructure

Chapter 1. Introduction

1.1 Overview

The Internet has opened unprecedented opportunities for business collaboration among corporate organizations tailored to share common automated services and resources, but not without cost. Business transactions now cut across geo-political boundaries with and without pre-existing trust relationships aimed at providing competitive offering to both consumers and supplier of services. The emergence of Web Services platform [1] which is a promising vehicle for dynamic composition of services and offerings is revolutionizing global transactions in an unanticipated way. Yet, these new ways of service provision are raising fresh security threats including privacy, confidentiality and trust [2, 3]. Access control [4-7] has emerged as a popular solution for treating application level security in the context of resources control against unauthorized access.

Trends in emerging access management systems raise an interesting paradox. On the one hand, service providers' applications require identity/attribute related information in order to validate a user's request. On the other hand, users may not wish to disclose their information or attributes to a remote Service Provider (SP) without determining in advance whether the service provider can be trusted to comply with their privacy preferences. In spite of this, current access control and privacy protection systems are typically unilateral and provider-centric, in that the enterprise service provider assigns the access rights, makes the access control decisions, and determines the privacy policy. However, in closed systems [8, 9] where a sort of pre-knowledge exists among communicating parties, security or privacy threats are of less concern than in open systems comprising several parties who may not share any pre-existing knowledge or trust relationships. Consequently, privacy is a growing concern, and has raised a number of social, cultural, legal and technical issues which make the protection of privacy across security realms more challenging.

The quest for efficient privacy measures has received enormous inputs from both research communities [10-12], privacy protection advocates [13-15] and technical standards [16, 17]. Access control is the mechanism for constraining access to sensitive resources. In high-level security environments i.e. those in which the risks involved in business transactions are high, this usually takes place in two phases, namely authentication and authorization [18]. Whilst authentication is the process of verifying and validating the identity claim of an entity attempting to perform some action, such as accessing a network or partaking in a transaction, authorization deals with the process of determining whether this authenticated entity has the privilege to access the resources under given local conditions and rules. Both phases imply the use of personal identifiable information (PII) that is likely to be traceable to a communicating entity.

It is important to state that an individual is associated with a diverse set of attributes and/or properties, which may be used in access control operations, and have the potential of being correlated. That is, the individuality is connected with a range of attributes, or properties including driver's license, postal address, national insurance number, marriage certificates, email addresses, telephone numbers, membership certificates, vehicle plate numbers, job functions, affiliations, etc. Additionally, individual activities can generate information that can be traceable e.g. when an individual purchases an item online, and is given a proof of purchase, e.g. a receipt, this can later be linked to that individual and used in a manner that is undesirable.

1.2 Background

Privacy as a socio-economic problem has impacts on individuals, communities, governments and businesses. It is burdened by legal, social, behavioural, psychological and cultural factors, which are subject to numerous expectations and interpretations [19]. These expectations are affected by certain compelling forces including economic benefits, enterprise business interests [20], regulation, legislation [21] and advances in technology [22]. These raise the question of how realistically we can expect privacy protection, given increasingly varying digital environments. These forces when viewed from a socio-economic perspective explain why finding widely acceptable privacy violation countermeasures is becoming increasingly difficult [23]. The fact that privacy is often perceived from dissimilar cultural backgrounds, resulting in an array of unfulfilled expectations and trust levels, further complicates the privacy problem space.

Furthermore, many individuals' poor understanding of privacy and its importance [10], plus the fact that emerging activities that have the potential to violate privacy are sophisticated, further erode the user's ability to participate in privacy self governance. In most cases, users are not fully aware of the information held about them, where the information is stored, or the risks of undesirable use. Acquisti et al argue that a considerable number of individuals may be willing to exchange privacy for convenience or agree to release PII in return for small benefits or rewards [19]. For example, a credit card company could make an insignificant offer (e.g. slightly reduced interest rate) to lure individuals into giving PII. Another example is where an online survey company may give out vouchers to individuals that are willing to participate in a marketing survey. Moreover, empirical study [20] has shown that users are not even sufficiently informed to make important decisions or risks assessment concerning their privacy. Acquisti et al in [20] stated three main challenges faced by an individual's privacy decision making process as:

- Incomplete information caused by external factors: Third party sharing of information in a transaction without the individual's consent or even when the individual is not part of the transaction;
- Information asymmetries: Information that is applicable to the privacy decision process may be available to only a subset of the parties involved in making the decision;
- Individual innate bounded rationality: Individual's inability to sensibly process a large amount of information in depth prior to making a decision.

In [19], they also argue that "subjective perceptions of threats and potential damage, psychological needs, and actual personal economic returns" affect individuals' privacy decision making process. Nevertheless, the individual is faced with privacy realities and expectations, and any attempt to view privacy from an extreme standpoint could mean secrecy, anonymity and solitude. The implication is that any move to attain absolute privacy could then result in total anonymity such as the absence of traceability, linkability and observerability of the privacy subject [10]. Of course, these would result in the inability to conduct business where personal information is needed in order to process and complete business transactions.

Privacy has also raised a number of fundamental questions [24, 25] regarding so called personal data: i.e. which information can be classified as personal? Who should be the custodian of it? Who should control access to it and under what conditions? What is the context under which personal data should be accessed, i.e. for what purposes and what should happen afterwards? Other issues of concern include whether the techniques used during collection, processing and use are by any means legal. Furthermore, other considerations comprise whether the data owner is fully aware about the data and the processing- consent, notice and awareness, the quality of data, i.e. is the data accurate and a true representation of the data subjects' attributes or properties?

Though some researchers and various privacy standards and principles [13, 14, 26] have been attempting to address these questions, development of efficient technical solutions that address the privacy concerns, remains a challenge that has continued to attract research inputs [27]. Another critical privacy factor is the inherent conflict between privacy protecting PII on the one hand, and the legitimate free flow of information on the other. This buttresses the point that absolute privacy will pose undue interference with the legitimate free flow of information [28]. Bearing these factors in mind, it can be argued that privacy protection can be viewed from two angles: the need to decouple identifying information from real world identities whenever practicable, and where it is impractical, the appropriate enforcement of privacy measures based on the individual's security preferences.

At this point, it is useful to clarify more precisely what we mean by PII. To this end, the following definitions are offered:

i. Personal Identifying Information is information that may be used to establish the identity of an individual (or an entity).

ii. Personal Identifiable Information is information that may be associated with an identifiable individual (or an entity).

iii. Personal Identifiable Information is information that may be used in a manner that causes harm to an individual with whom it is associated.

iv. Personal Identifying Information is any digital content the unauthorized disclosure of which may raise privacy and confidentiality concerns.

In this thesis, the term PII is used in respect of definition 2 above. In other words, PII refers to any piece of information associated with appearance, personality, knowledge and characteristics that describe an individual (or an entity) [29].

Several attempts have been made to address privacy concerns especially when computer databases became widely used. In recent times, many of the privacy initiatives focus on website privacy activities, that is, purely on how websites collect, share PII and what they promise to do with PII. So far, privacy concerns have not received the required attention in distributed access control infrastructure. In cases where privacy issues are addressed, privacy is considered from the client's perspective and confidentiality from the service provider's. But at both ends, clients and service providers [30] require to determine in advance the assurance that the other party can act in a manner that is compatible with their own security preferences.

Nonetheless, the distinction between preserving the privacy of service outputs and the confidentiality of service meta-information (identities, attributes, policies etc.) is becoming blurred [31]. The underlying assumptions have led to the development of solutions that are typically unilateral and asymmetric. In contrast, emerging access control requirements point to a need for a bilateral paradigm, simply because both the server and client entities may have privacy and confidentiality concerns. This has resulted in a new research area otherwise known as trust access management systems which employ trust negotiation concepts whereby communicating parties can reciprocally release access control policies and credentials [8, 32, 33].

Furthermore, growing business requirements may demand the dynamic exchange of service requirements, contractual and service level agreements in order to assess the mutual benefits and associated risks before engaging in high-risk business transactions. Enabling the runtime exchange of these business requirements requires a bilateral and symmetric infrastructure that will allow communicating peers to indicate their willingness to accept constraints imposed by the other party, before the latter is prepared to reveal their sensitive information. There is some overlap between users' privacy requirements and business requirements, meaning that privacy, confidentiality and trust are typically associated.

Addressing confidentiality and privacy problems mutually and simultaneously, requires that parties in distributed transactions should have a uniform way of declaring their security requirements alongside the constraints they may impose on the use of their information before sharing it. This provision will ensure that parties evaluate the risks associated with giving out their information and determine the degree to which they are prepared to trust other participating parties. This entails the requirement for communicating parties to identify constraints and obligations they may wish to place on the others concerning the use of their resources.

1.3 Current Technological Situation

A variety of solutions have been proposed in an attempt to solve the problems imposed by privacy and security concerns including anonymous schemes [34], the W3C Platform for Privacy Preference (P3P) [16, 35], Enterprise Privacy Authorization Language (EPAL) [24, 36], amongst others; and they have had impact in reducing privacy risks. However, most of these efforts focused primarily on website privacy and less in access control operations. Equally, where privacy is considered in existing access control solutions, it falls short of basic principles of privacy. As a consequence, the investigation of privacy options is approached in a more pragmatic way focusing on the development of a viable system that addresses privacy and confidentiality with the capability to ensure remote enforcement of security constraints. More importantly, the outcome of this work technically speaking is a model that has the potential to merge mechanical and socio-legal privacy components into a common framework. The following summarizes the characteristics of existing systems:

- Most existing attribute-based access control systems are unilateral: the service user (client) is required to disclose PII unconditionally to the service (service provider application), irrespective of whether the service can adhere to the security preferences of the user, otherwise access to the service cannot be allowed.

- The mechanism for privacy assurance is overlooked. The client cannot be confident that the information passed to another party will be well protected and used only by the authorized parties and for the intended purposes. The assumption here is the fact that the attribute consuming party does not sign any obligating constraints before the collection of personal data.

- Given that preserving privacy may demand the fulfilment of contractual obligations e.g. compliance with foundational privacy principles, such as Fair Information Practice (FIP)[14], current systems have not provided effective means to allow a client to participate actively in the governance of attribute-information.

- Current systems fail to consider server side privacy. On the other hand, service providers may have certain information that the client would like to see (i.e. policy, metadata, etc) disclosure of which to arbitrary strangers may leak important business information.

- Traditionally, an access management system involves a so-called trusted third party (TTP); the privacy of third party affiliates is disregarded at present. The consequence is that the arbitrary disclosure of trust relationships with a TTP may sometimes violate important business rules.

These characteristics establish the need for further investigation of access control infrastructure with a particular focus on privacy. The bilateral treatment of privacy, confidentiality and trust is critical for privacy assurance. Based on this assumption, the focus was on access control frameworks that can be derived from conventional concepts [8, 33, 37] and standards [37, 38], and which potentially will enable bilateral privacy negotiation between a service and a client. This led to the development of the concept of Obligation of Trust (OoT) [31], whereby two parties can exchange difficult-to-repudiate[1] digitally signed business constraints and proof of acceptance.

1.4 Research Motivation

The explosion of the Internet resulted in unprecedented manipulation of PII in terms of collection, storage and dissemination [21, 39]. As a result, the level of identity fraud has continued to increase dramatically, which leaves ordinary users disadvantaged [40-42]. In the UK, a Cabinet Office Study in 2002 indicated a rise in the use of false and stolen identities, and estimated that crime caused by identity fraud may cost the UK about £1.3 billion per year [40]. Arguably, in many cases, the violations of privacy are economically and politically encouraged, and often, the individual is trapped into them naively due to economic or other incentives [21, 23]. In the commencement of the study, the author participated actively in the

[1] We use the term "difficult-to-repudiate" rather than non-repudiation, since repudiation is a legal issue that has to be determined in a court of law. The technical constructs proposed in this thesis should make it more difficult for an entity to repudiate their actions.

design and development of an access control policy editor, an empirical usability evaluation of which was published in [43, 44]; thus exposing him to access control issues. The motivation to undertake this work came from the quest to investigate privacy problems, which if uncontrolled may potentially outweigh the benefits of Internet computing. This work therefore intends to provide a novel interface that could significantly contribute to addressing privacy problems, and make a substantive contribution to the body of knowledge in the area of privacy and security.

1.5 Research Aims and Objectives

The principal aim of this study is to develop a means to allow parties in business transactions to participate actively in privacy preservation in access control infrastructures. In order to address this, it is necessary to undertake an empirical investigation of the options for addressing privacy concerns in access management operations. This involved a thorough investigation and analysis of the privacy problem space, especially in the context of open environments where business transactions are likely to involve parties with or without pre-existing trust relationships. The main objectives of the study are outlined below:

- Examine and understand the privacy space, confidentiality and trust concerns in open systems through formal investigation of related security and privacy concerns;

- Investigate existing systems in the context of privacy protection to expose the gaps if any, highlight their benefits, strengths and weaknesses, and develop a viable solution to fill the gaps;

- Examine trust in the context of privacy across the boundaries of security domains from the standpoint of dynamic trust establishment;

- Examine how existing Internet standards such as XACML, SAML, etc, and privacy enhancement technologies can be utilized in addressing the privacy problem, and if necessary extend such standards to support the technical solutions;

- Develop a framework that has the potential to address some of the privacy problems with a software implementation as a proof concept;

- Critically appraise the work with respect to successes and substantive contributions to the body of knowledge in the field of privacy and security.

The final deliverable is an infrastructure that is capable of addressing the privacy problems uncovered by this study.

1.6 Research Questions

The main question addressed by this work is "how can privacy be preserved when a communicating party's personal information is involved in access control operations"? In other words, can the recipient party (or system) be made to treat a party's PII in a manner that respects privacy? This main question is further broken into the following:

1. How can a party be made to protect the privacy and confidentiality of personal information provided to it?
2. How can a communicating party participate in decisions regarding protection of privacy and confidentiality in access control operations?
3. How can privacy be guaranteed across security domains of trust?
4. How effectively can privacy guarantees be enforced across autonomous security boundaries?
5. How can individual security preferences be balanced against the legitimate free flow of information, given that a party's privacy may sometimes clash with national, socio-economic or business interests?

1.7 Key Contributions

The thesis, through the research process followed in seeking to answer the above research questions establishes new knowledge in the areas of privacy and security, and particularly in the infrastructure required to support privacy and security. The knowledge gained is viewed from two angles: theoretical and practical.

In theoretical terms, the study provides:

- detailed review and analysis of the privacy and security problem space;

- critical comparative analysis of current access control systems with respect to privacy;
- new insight into the relationships between privacy, confidentiality and trust in addressing a common problem;
- exposure of the unrealistic assumption about privacy and confidentiality in server-client architectures, as well as how trust is related to both;
- support for a symmetrical architecture, which allows both service consumers and providers to mutually negotiate for the release of their resources and properties that raise privacy concerns;
- supplementary substantiation in the literature that too much dependence on transitive trust provided by trusted third parties can no longer sufficiently protect privacy and confidentiality in high-risk business environments;
- contributions to refereed privacy and security literature [30, 31, 43, 45-47].

In practical terms:

- a conceptual design and development of a technical infrastructure that can automate the provision of 'hard-to-repudiate evidence[2]' i.e. a digitally signed acceptance of obligations (constraints imposed by a party) that can possibly be used in a court of law should a dispute arise, and that can be automatically processed by the obligating party;
- the product of this work is a significant deviation from traditional access control systems, but is based on well-known standards, making the approach pragmatic and adaptable;
- the software implementation demonstrates a proof of concept, and the potential applicability of the framework in real-world scenarios.

[2] The term evidence refers to a piece of information that is critical to convincing management or legal authorities that some kind of a breach has occurred.

1.8 Structure of the Thesis

Research is an extensive study that requires a proven methodology, and systematic approaches underpinning the rigorous investigation, analyses and appraisal of an entire study. The design of an appropriate paradigm and the understanding of its philosophical underpinnings are critical elements in the success of a research endeavour. Chapter 2 discusses this work's research methodology.

Privacy and confidentiality are key challenges facing ubiquitous computing environments and the Internet in particular. Privacy concerns have raised a number of legal and regulatory issues, legislation and regional guidelines; and so this requires an understanding and assessment of foundational privacy principles [48]. Chapter 3 looks into privacy concepts from the perspective of privacy laws, standards, legislation, regulations, and principles, and assesses them in the context of this work.

Access control mechanisms are important security services that can deal with application level security and privacy concerns. However, many access control systems make extensive use of PII to determine the access privileges of system resource users. The use of PII in access control operations raises key privacy and security concerns, particularly in distributed environments with different autonomous security domains. Chapter 4 discusses foundational elements of access control, examines privacy characteristics of current systems in the context of distributed environment. The primary goal is to expose their strengths and weakness, and presents a critical analysis of these systems in the context of the privacy problem space.

The advent of computer and communication networks changed the landscape of human interactions with others, resulting in the need to make information unreadable to unintended recipients. In order to conceal information from unintended recipients, extensive studies have been undertaken to find ways to make information opaque to others. Chapter 5 takes a look into the fundamentals of information security focusing on technologies that help to alleviate privacy and confidentiality problems. These technologies are considered very relevant in the development of a viable technical infrastructure that can support privacy and confidentiality protection in distributed environments.

Security Standards provide the platform for promoting application service interoperability. Utilizing standardized tools that have been reviewed by a large community of experts and users is considered as a way to ensure that the resultant utility of his work can be adapted in real-world scenarios. Chapter 6 explains the basic concepts and key features of the specific standards, components and architectures that significantly influenced the study. In particular, the XACML and SAML standards are described and analyzed in more details.

One of the expected outcomes of this work, besides new contributions to the knowledge, is the development of an artefact that demonstrates the applicability of the basic concepts discussed. Given the multiple level of interactions between participating parties in distributed access control operations, a requirements gathering and analysis are critical to the development of the proposed technical framework. The systematic development of an artefact is an important design-research strategy, which can substantiate the relevance of a study in the context of research work. In Chapter 7 an XACML Trust Management Infrastructure is proposed to incorporate privacy and trust into the XACML access control framework. It describes a security threat modelling, which helps to determine the key privacy vulnerabilities and security threats. The idea is to validate earlier assumptions made and provide an interface for the concrete implementation of a proof of concept.

One fundamental challenge faced by this study is how to make parties in autonomous security domains treat PII with respect to privacy. The problem is intrinsic due to the open characteristics of the underlying communication networks that provide the backbone for the interactions. In the progression of this work, the notion of Obligation of Trust protocol was conceptualized in an attempt to address the remote enforcement of privacy obligations. Chapter 8 gives details of the concepts that underline its protocol and message exchanges.

The implementation of a software as a proof of concept that demonstrates the applicability and capabilities of the proposed technical framework was considered essential. Chapter 9 describes that systematic implementation with a detailed technical design of XTMAI system architecture and core building blocks, plus an illustration of how the system works.

A formal evaluation of a research study is a critical success factor, which provides a feedback whether the work met its stated objectives in terms of the quality of the design process. Chapter 10 presents the critical appraisal of this work in the context of the research questions and selected current systems.

Chapter 11 concludes this thesis with highlights of the benefits, strengths and limitations of the resultant framework plus suggestions for further and future work.

1.9 Conclusion

The Internet, which has provided the backbone for unprecedented business collaboration among parties, is faced with a range of security problems including privacy and confidentiality. Most existing access control systems designed to restrict access to sensitive resources overlooked these aspects from client's perspective. This chapter has introduced the background materials in the subject area, overviewed the privacy problem space, plus the indication of current status of the privacy protection capabilities of existing access control solutions in distributed environments. The motivation for and objectives of this work, and the questions addressed by it have been thoroughly dealt with in this chapter. Finally, key contributions to the body of knowledge presented in this thesis and an overview of the rest of the chapters of it are also highlighted.

Chapter 2. Research Methodology

2.1 Introduction

In [49, 50], methodology is described as the "the analysis of the principles of methods, rules, and postulates" or "a collection of theories, concepts or ideas" that underpins a particular field of study. Oates [51], defines a methodology as a "combination of research strategies and data generation methods" employed in the research work, and argues that it is essential to differentiate research work from normal project-oriented development. Research methodology is an intricate aspect of any research work, partly because of the diversity of the philosophical underpinnings, but essentially as research involves rigorous processes with many assumptions that can affect the applicability of the results.

This chapter discusses a set of approaches and techniques that underpin this work, the rationale for the design decisions taken and the various analyses carried out. The philosophical approach [52] that justifies a thorough research activity is also presented.

2.2 Philosophical Approach

This work relates to a socio-economic problem which impacts on individuals, communities, governments and businesses. This problem is intrinsic to the relationships and interactions between these parties, and its solution involves trade-offs between competing interests, and individual perceptions. Because of this, it is important to consider the philosophical underpinnings of this research. In a research community, situating a research study based on its philosophical stance is often one of the most intriguing challenges faced by a researcher. This is simply because a research work can cut across several domains of knowledge and various disciplines. The Positivist, Interpretive and Critical epistemologies [53] dominate contemporary social research. The positivist assumptions are based on the fact that empirical scientific techniques suggest that reality is a function of objectivity and can be measured independent of the observer [54, 55]. The interpretive approach is based on the assumption that access to reality is through social constructs and that the understanding of a phenomenon can be through 'people's perception' i.e. people assigning meaning to an event. On the other hand, the criticalist's assumption is that social reality is historically dependent, and so can be produced by people [53]. However, in [54], Lee argued that these epistemological approaches are philosophically distinct, but in reality their distinctions are blurred.

Science has been viewed by many as a social phenomenon and a problem-solving undertaking that underscores its assumptions on theories whose predictions can be experimentally disproved or proved [55]. Arguably, this study relates to a socio-economic problem that has multiple effects (i.e. socio-economic, political and cultural impacts) on individuals, governments, and businesses. This study falls under the positivist epistemology, it is a problem-solving process [52], which is about developing a technical solution that aims to address a socio-economic problem concerned with privacy violations. The understanding of the issues and variables (characteristics) inherent in developing a suitable technical framework to address the problem requires extensive research activity.

Therefore, this study aims to develop a technical framework that supports communicating parties in business transactions by allowing them to participate actively in decision making with respect to privacy protection in access control infrastructures. In summary, therefore, this work can be classed as a process concerned with the creation of suitable technology to aid the resolution of a socio-economic problem, which can result in the production of an artefact [51, 52]. In [51, 56, 57], this is classified as design and creation research, whilst in [52] it is referred to as design-science research. According to Hevner et al [52], IT artefacts are "*constructs* (vocabulary and symbols), *models* (abstractions and representations), *methods* (algorithms and practices), and *instantiations* (implemented and prototype systems)". They argue that the result of the design-science research is "a purposeful IT artifact created to address an important organizational problem". Overall, the purpose of this work is to address privacy concerns through the "building and evaluation" [52] of a viable technical framework.

2.3 Overview of the Research Design

This work is divided into two logical segments, namely investigation of the problem space and the development of a technical solution to address the problem considered.

- Investigation of the problem space: This segment deals with the empirical investigation and analysis of the problem space without too much emphasis on the technological underpinnings and implementation. The idea is to capture the problem domain, and understand the conceptual variables and complexity without reference to any implementation details. Chapters 1, 3 and 4 cover this segment.

- Technical solution: This stage is concerned with the technical requirements, systems analysis, architectural model and a software implementation as a proof of concept of this work. Chapters 5, 6, 7, 8 and 9 cover this aspect of the work.

In both phases, the approaches described in [51, 58] were followed; and figure 2.1 is the conceptual research design that shows the overall steps taken in conducting this study, and they are described in the sections that follow.

2.3.1 Awareness of Problem

According to [51], this stage is the recognition and identification of the problem domain in terms of the main issues of concern. There are several social and technical issues that make privacy problems a growing concern. Thus there is a need for a deepening understanding of the characteristics of these, and they are essential in developing potential technical solutions that could address the privacy concerns. In the awareness phase, the sources of materials include literature reviews of relevant work, various Internet standards, technical documents, and concrete access control implementations, in an attempt to identifying their strengths, limitations, and gaps in existing knowledge. Additionally, it is important to identify some of the underlying assumptions underpinning existing systems and their approaches. The rationale is to develop an in-depth understanding of the socio-economic elements of privacy and security in the context of access control operations in distributed environments. Chapters 3, 4 and 5 cover this segment of the thesis.

2.3.2 Suggestion

This phase is concerned with the development of the research motivation, the objectives and main questions addressed by his work. The deductions from the awareness of the problem, serve as an input to this phase. In the process, it may be essential to examine further the outcome of the awareness stage with a view to analyzing the underpinning theories and assumptions identified in order to put the generated ideas into proper context. It may require further investigation of relevant literature and analysis of the current systems. While chapters 3 and 4 served as inputs to this phase, chapter 1 deals with this aspect of the thesis.

2.3.3 Development

This phase may be called the "build-and-evaluate loop" [52], and is inherently iterative. It is concerned with the requirement analysis, the architectural model design, and the implementation of the software system of the proposed technical solution. The technical requirements are analyzed based on a security threat modelling technique [59] discussed in chapter 7. The purpose of the security threat modelling is to capture the privacy and confidentiality requirements in terms of vulnerabilities and associated threats from which architectural model design can be drawn. The modelling helps to deconstruct the interactions among the actors involved in distributed access control operations, and to determine how effective trust relationships can be established among them to guarantee privacy and confidentiality protection.

Chapters 5 and 6 provide useful inputs in conceptualizing the design of the framework. The development stage also involves the design and modelling of the system components using Unified Modeling Language (UML) techniques [60] to show object classes and relationships

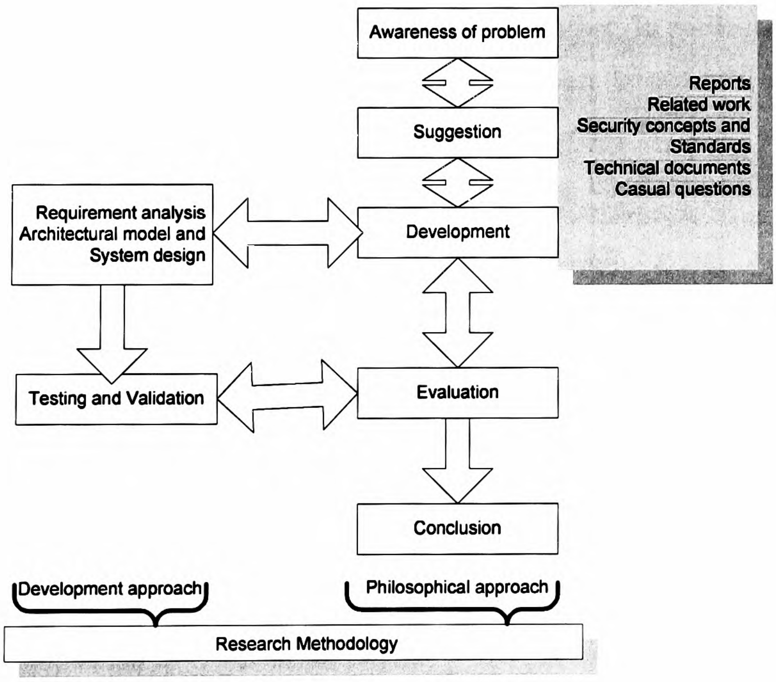


Figure 2.1 Abstract view of the Research Design

among them. As shown in figure 2.1, in this, revisiting of previous stages might be necessary to gradually modify and improve on the solution considered. Chapters 8 and 9 provide more technical details of this segment of the thesis. The techniques employed in this phase are detailed below.

2.3.3.1 Developmental Issues

The first thing considered is the various technical options and approaches that might be adopted in addressing the main research question. There are several approaches to addressing privacy and confidentiality concerns including anonymous credential systems [61] [62], access control systems [63, 64], etc. Anonymous schemes can support the protection of users' privacy and confidentiality but in environments in which transactional risks are high, and tangible credentials are required before access is granted, anonymous solutions can make accountability and non-repudiation of privacy invasions difficult. In contrast, policy driven access control schemes can support the protection of users' privacy and confidentiality, and provide mechanisms that allow accountability and non-repudiation, but can be vulnerable to abuse and/or misuse. These options are considered in the context of the problem domain.

Second, other issues considered include the availability of privacy enhancement technologies and standards, and how they can fit into the technical solution. In particular, the choice of an access control model that is flexible and extendable to support the potential solution is a critical design decision to be made. When considering the programming language and platform to use, the availability of an open source Application Programming Interface (API) that could support the software implementation was primarily considered.

2.3.3.2 Requirement Analysis

Traditionally, the software engineering process is independent of the application area, project size or complexity and primarily consists of three key concepts: the "what"? (Problem definition phase), the "how"? (design and development phase) and "support" which focuses on the changes that may be necessary after development [65]. The definition phase captures all the what questions: the information to be processed (i.e. privacy and confidentiality concerns to be addressed), requirements in terms of functionality and performance issues, system behaviour and interfaces, design constraints, evaluation and verification criteria. This phase focuses on two basic steps: requirements analysis and conceptual design (presented in chapters 7). The purpose is to deconstruct the different actors in distributed access control operations, which is critical in understanding of the risks involved in the exchange of attribute information. In addition, the analysis of technical issues surrounding the building of trust relationships between communicating parties from different autonomous security domains will dictate whether a given trust relationship model is suitable and appropriate in the environment under consideration.

2.3.3.3 Implementation Approach

The Software Development Life Cycle (SDLC) is considered, which is a proven process of software development that has the benefit of structured planning and control [56]. This generic model provides the platform for using other techniques such as Object Oriented Programming (OOP) in the application architecture and UML in modelling the object classes. These techniques help in determining the system components' dependencies and relationships. Project based developments usually follow stepwise methods such as the waterfall lifecycle [56, 65]. In contrast, research based developments favour rapid application development (RAD) paradigms, which are more iterative, and allow refinement in the entire development process. The RADs are essentially good where all technical requirements cannot be determined beforehand [66] such as in research environments. According to [51], requirements can be refined as a result of further analysis of problem space and recycling of the development process becomes imperative. According to [66], the respectability of prototype paradigms is as a result of a proven track record of dynamic responses to changes in user requirements, which reduces the amount of reworking needed and has helped to control the risks of incomplete or inaccurate requirements. The rapid prototyping paradigm is considered for the software implementation. Besides the general notion that rapid prototyping is suitable for research projects [51], the rationale for the choice is complemented by the author's previous experience [43].

2.3.3.4 Software Debugging

Two approaches are used in the debugging phase. First, a repeatable incremental unit test technique that can detect errors at the earlier stage of the process. The objective is to significantly reduce the cost of reworking, minimize system failures and reduce the amount of potential bugs. Rapid prototyping is found useful in this context, it allows incremental and unit testing of unit blocks aimed at verifying that the codes actually perform what they are designed to do. In a software lab environment, sandbox-testing tools would give rapid and more meaningful results in both testing the functional and non-functional requirements. However, in a research environment sandbox tools are unavailable, which usually prompts the use of in-built debugging tools within the software Integrated Development Environment (IDE), which in this case is Netbeans 6.1 [67].

2.3.3.5 Technical Testing and Evaluation

In this segment, a range of technical tests and evaluations to validate the software against the assessment of the main research aim was planned. Although traditional software testing techniques such as the unit test are employed during the coding process, object classes and method [56], testing of the software in terms of functionality and performance is important. Two main types of validation, namely technical functionality and performance are chosen as rationale to justify the capability and applicability of the product of this work. The evaluation is done in two steps. First is an assessment of how the developed prototype addressed the main research challenge in view of its requirements. Another segment is a performance test such as a throughput test to determine the average response-time of a typical negotiation between communicating parties in distributed access control operations.

2.3.4 Evaluation

The evaluation stage is critical and should have some measurable metrics. Hevner et al [52], and Oates [51], stated that a design research which resulted in an artefact can be evaluated in terms of functionality, completeness, performance, accuracy, usability, etc, but has to fit into the original objective of the research. The evaluation compares the solutions against the objectives and research questions. In line with the above, three metrics are considered upon which to evaluate the resultant technical framework. First, technical validation seeks to test the functionality against the objectives and research questions using experimental and simulation approaches. Second, performance evaluation seeks to measure the average response-time during the privacy negotiations. This particular evaluation hinges on many factors i.e. network characteristics, volume of computation depending on the number of rounds and access control input data. Third, carryout a comparative assessment of the proposed solution against some of the work reviewed in chapter 4. Chapter 10 covers the evaluation phase.

2.3.5 Conclusion of Study

This segment concludes the study and gives a summary of the research outcomes; draw useful conclusions based on the research successes, knowledge gained and the known limitations of the work. In addition, key contributions to the body of knowledge in the area of privacy and security, and useful suggestions for future work will be presented.

2.4 Conclusion

The need to discuss the methodology that underpins a research work is essential to substantiate its rigorous processes. This chapter discussed the philosophical and methodological underpinnings of the work described by this thesis, the system development paradigms, methods and techniques employed in developing the resulting framework. Additionally, the rationale that justifies some of the choices made in the design of the technical framework and software implementation has also been presented.

Chapter 3. Review of Concepts of Privacy

3.1 Introduction

Extensive studies have shown that privacy related issues are growing in the current ubiquitous network environments based on the Internet, and are receiving widespread investigation within the research community [11, 23, 68, 69]. In spite of the fact that the Internet has had significant visible impacts in enhancing distributed transactions with exceptional economic benefits to individuals and businesses, the rising security threats and challenges are disquieting [70]. Although various efforts in advancing security techniques to deal with these threats are partially successful, the level of sophistication of new technologies continues to evolve significant new threats [22].

The failure of existing solutions to effectively tackle these challenges cannot completely be ascribed to system failures; fresh business requirements demand new technologies, which in turn also raise new security issues [71]. This is worse in a heterogeneous environment where transactions span multiple autonomous security realms and are unsafe. Moreover, complexities in the gathering, processing and sharing of personal information are changing the landscape of these Internet security problems. Privacy is a critical concern, and has generated unprecedented debate, perhaps due to the complexity and ambiguity of its notion coupled with the fact that it lacks an agreed definition [23]. This complexity, plus the fact that it is only vaguely understood, hampers any effort to find a common solution [23]. As complementary efforts, privacy laws, legislation and standards have been enacted by regional institutions and governments to help in reducing the impact privacy challenges impose on the network-centric global economy.

This chapter examines the foundational privacy problem space, and reviews the concept of anonymity with emphasis on its strengths and limitations. In addition, a few of the privacy protection initiatives and principles in some economic regions including the United State and Europe are examined. Finally, a comparative analysis of these privacy principles is given to broaden the understanding of privacy protection requirements in the context of access control system infrastructure.

3.2 A Survey of Privacy Problems

The first mention of privacy is generally accredited to Samuel D. Warren and Louis D. Brandeis in their famous paper: The Right to Privacy, in which they defined privacy as "the right to be left alone" [72]. Alan Westin [73] of Columbia University expanded on the definition of privacy as "the claim of individuals, groups, or institutions to determine for themselves when, how and to what extent information about them is communicated to others". Arguably, Westin's definition somehow influenced the general perception of the term privacy, and to a certain extent may have driven many privacy standards and initiatives. The dawn of the Internet, and the increasing reliance on network based business transactions opened up a separate new socio-economic dimension that impinged on privacy, confidentiality and trust. This demanded more investigation of the privacy problem domain. Privacy as a concept is affected by a variety of socio-economic factors, including business interests, regulation, legal requirements and the advancement of technology [34]. The individual needs to transact business with others, and this requires exposure of personal information. Businesses want to offer customized services and this requires personal information and profiles [74]. Governments want to ensure security and require some aspects of strategic personal information. New technologies that promise better user experience raise fresh privacy dimensions. For example, the Federated Identity Management(FIM) scheme which gives a Single Sign On (SSO) experience, i.e. the use of several application resources without the requirement to logon multiple times has exposed new privacy risks [75]. Simply put, FIM has the potential to expose PII to arbitrary strangers [70, 76, 77].

The incidence of privacy breaches such as identity theft, unsolicited offers, phishing attacks, and probing attacks with their far reaching costly consequences continues to increase dramatically [76]. In the UK, a Cabinet Office Study in 2002 revealed the rise in the use of false and stolen identities [40]. According to [78], the US Federal Trade Commission (FTC) received 255,565 consumer complaints in 2005 concerning identity theft cases. A report by the Identity Theft Resource Center in the US revealed the impact of privacy breaches on their victims, and the fact that a significant number of the victims bear the liabilities for such crimes [79]. In the US, the dominant threat to personal information is data brokering, which involves gathering, processing and dispersal of personal information by third parties, and is vulnerable to various misuse and abuse. In effect, data brokers who sell personal information have unconstrained access to many databases where personal information is stored [76]. The implication is that there are concerns about the undesirable use of personal information; whether it is obtained by lawful means or not is not the issue. Moreover, a study published in 1999 by AT& T Research [80] revealed that about 87% of users were "very or somewhat concerned about threats to personal privacy while online". But, in [20], Acquisti et al, state that an individual's behaviour with respect to privacy decision making can be affected by social preferences, norms or cultures. This implies that the factors that can contribute to the inability of an individual to make privacy sensitive decisions may not be explicitly controlled by the individual.

Therefore, the results of identity disclosures, linking data traffic to identity, location disclosures, user profiling and the undesirable disclosure of identity information to third parties give rise to a multiplicity of privacy concerns. The quest for privacy is not unconnected with public craving and individual yearning for proper data governance. Individuals are increasingly vulnerable to privacy threats because of the radical change in information sharing mechanisms driven by the Internet [22, 70]. Companies have automated their data collection mechanisms allegedly to provide their customers with customized services, value offers, competitive offerings, etc [22]. The pretence upon which privacy violators amass huge amounts of personal identifiable information from network entities which they can employ to create detailed profiles for aggressive marketing campaigns (on the side of commercial entities), fraud (in the cases of identity thieves), and national security countermeasures (in the case of government agencies) [21, 76] is disturbing and unsafe. Incidents of privacy violations are frequently reported in the media [21, 81, 82]. It is inevitable that as entities use public networks, they leave traces of information at different nodes or sites, which can probably be collected and analyzed by interested parties.

The above insight establishes that privacy activities are complex and sophisticated, and require several aspects including technologies, legislation, laws and guidelines to be examined. The underlying disclosures of privacy concerns raise the question of how much privacy protection we can realistically expect given the inherent characteristics of open environments. Several organizations have distinct and often conflicting security requirements and expectations which complicate privacy solutions. The issue of establishing sufficient trust to screen access requests effectively, for both internal and external users, implies significant technical tasks to be carried out. Individuals are concerned with the information that they pass to others, and yet they want to reveal information that can facilitate interaction. Consequently, privacy is a concern to everyone due to the misuse and/or abuse of PII, which in most cases leaves the victim disadvantaged. On many occasions the individual is made to bear the risks associated with privacy invasion. These risks may be in the form described in [3, 20, 23] and are summarized as follows:

- financial loss due to stolen identity or interaction with imposters, (i.e. a phishing attack tricks its victims into revealing their credentials such as online banking authentication details. The attacker can use a victim's credentials to make undesirable financial transactions on the internet);
- the unintended release of identity attributes may impact on the individual's financial credit status;
- long term consequences of misused credential attributes that can significantly disrupt an individual's social life and status;
- discrimination such as social services exclusions [20], economic and political alienation.

The enforcement of privacy across multiple domains is intrinsically problematic [30, 31]. The gains achievable from distributed transactions cannot be fully realized without addressing privacy barriers to them [11, 14, 20, 23, 25]. Moreover, the need arises to consider the impact of privacy principles and legal frameworks as they relate to protecting privacy. Besides, there are interest groups putting forward anonymity as a means to solving privacy problems, this leans towards total privacy but is of course difficult to realize in the real world [33]. From Westin's notion of privacy plus the fact that privacy related activities are complex, it still debatable whether addressing privacy problems through technology alone is sufficient.

3.3 Anonymity as a Panacea for Privacy Violation

Anonymity is one solution to the problem of how to preserve privacy. Anonymity has a very long history predating the advent of computers and simply refers to the state of being unknown or unacknowledged. In network communication, it refers to a subject using a computer resource or service without disclosing their identity or being linked to the usage [83]. Law enforcement agencies have used anonymous sources for both crime prevention and prosecution. People are sometimes willing to say something they don't want to put their signature to for fear of reprisal [84]. Historically, the use of anonymity in crime prevention invariably has inherent problems such as the credibility of sources and their reliability, and can sometimes be abused. Consequently, anonymity when not applied with caution can ruin the lives of innocent people. The properties of anonymity include untraceability, unobservability and unlinkability; and collectively they provide extreme privacy. The characteristic of unobservability is that a network entity can use a computing service without being observed by any party. Similarly, unlinkability refers to the notion that an entity cannot be linked to its use of computing services, and untraceability refers to the property that an entity uses services without creating traceable patterns.

Whether or not complete anonymity is desirable is debatable. In [85], Rowland suggests that measures that foster anonymity in order to protect privacy may, incidentally, facilitate or even encourage anti-social and illicit behaviour. However, anonymity has its proper use in privacy safeguards. There are several occasions where people have a legitimate need to remain private and unnoticed. People may have a high potential risk of being harassed, threatened or discriminated against upon the revelation of certain personal information such as medical records, political affiliation, etc. Another significant factor that encourages anonymity is the threat of price discrimination, which is very popular in online sales. In [74], Acquisti refers to price discrimination as the "seller's ability to provide the same commodity or service at different prices to different customers". The seller must have the capacity to track, trace and build considerable profiles of individuals in order to predict their purchasing capabilities if it is to implement price discrimination to its benefit.

3.3.1 Pseudonymity

The notion of pseudonymity is simply, establishing forms of transparent identifiers that cannot be directly linked to an identity. Pseudonymity has become an important privacy tool in recent times and is highly used in emerging Federated Access Management computer networks [86]. In distributed systems, a pseudonym possesses varying degrees of anonymity and is highly dependent on the existence of well-established vocabularies among communities of users. There are persistent identifiers that are traceable to users as well as single use identifiers, which completely conceal the user's identity but obviously reveal something about the user's origin.

The fundamental idea is that pseudonymity systems provide the means to shield real-world identities from being identified by undesirable parties but still provide mechanisms for giving account for the use of them. They offer a kind of 'trusted anonymous' identity i.e. a certified reference identifier, which when issued by an asserting party, can be recognized and trusted by the recipient party. The recipient can use it to determine whether or not to grant or deny access to resources. Essentially, the identity information of the user is not explicitly revealed to the recipient or relying party. The relying party trusts the identifier as a valid identity based on the trust it has established with the attesting party. This is the idea behind the popular SSO [86, 87], and its use is plausible where people can do business without the need to explicitly identify themselves at the service endpoints.

3.3.2 Pros and Cons of Anonymity

In some cases, anonymity is the failsafe to preserve privacy in the event that the relying party may not be trusted to preserve confidentiality. Consider, for example, a political activist who may trust the Inland Revenue to keep her PII confidential, but fears what a government security agency might do with her PII under the guise of national security. Various anonymity schemes [34] have attempted to address privacy problems in such cases. Though anonymity may be the only guaranteed option in certain situations, in many cases it is not a viable option [88]. There are cases where an entity must disclose personal identifiable attribute information in order to obtain a service or complete a transaction. This is the primary scope of this study; to investigate those cases requiring the confidentiality of information at the consuming endpoints. In particular, how to ensure privacy at remote site based on the security preferences of the providing entity. The main drawbacks of protecting privacy using anonymity are as follows:

- Anonymity has the ability to ruin the trustworthiness of an organization; typically, the underlying infrastructure could be misused or abused.
- Anonymity has the potential of being hijacked for illegal transactions, modification of sensitive data, spreading of virus and worms, hate speeches, threatening messages. etc.
- It can be susceptible to spreading of false information across the boundary of trust, which can be very damaging.

Anonymity may not provide privacy protection in many situations, particularly where other identity-attributes of a user may be required to complete access control based transactions.

Nevertheless, pseudonymity techniques are considered in this work, especially in distributed transactions where users must assess the risks of releasing their attribute information. In the context of this work, pseudonymity plays an important function in filtering out the service-requesting client, and providing a mutual way for communicating parties to establish initial trust context upon which higher trust can be established.

3.4 Privacy Legal Framework and Principles

The explosion in Internet transactions has escalated privacy concerns; organizations now consider privacy as a business requirement that must be fulfilled [89]. Privacy is a multi-disciplinary subject that spans technical, social, legal, and cultural issues, with an array of measures available aimed at protecting online privacy. Privacy is a rising socio-economic and technical problem that captures the interest of everyone using the ubiquitous networks and will continue to attract wider publicity resulting in more and more public awareness [21]. This has fuelled agitation for privacy safeguards and countermeasures to combat the impact of unregulated public networks on an individual's privacy. This quest has brought about the emergence of privacy laws, regulations, and legislation as well as fair information practices in most regions of the world. The varying privacy laws enacted worldwide are a bid to reduce the adverse effect of privacy violations by creating a privacy friendly environment that should consolidate the gains of the Internet. However, trends show that nations are also interested in balancing national security interests against privacy. This brings a debate about the roles the various privacy stakeholders can play amidst the requirements, dimensions and expectations of privacy [21, 89]. As a way of regularizing privacy enforcement, privacy standards have evolved in recent times not only to create the necessary awareness, but also to help a range of stakeholders to better estimate the risks associated with privacy activities [90]. The intense efforts by regulatory authorities, technical or business organizations centrally are in tune with the individual's increasing eagerness to be familiar with not only the privacy rights they can expect, but also what techniques or mechanisms are available to help enforce their privacy rights [20, 76]. The overall goal is to protect individuals with respect to the processing of their personal information.

Privacy laws and regulations emerged as safeguards to provide legislative and legal frameworks for treating information across organizations, regional boundaries, etc in terms of collection, storage, and sharing of information that links an identity. Empirically, these privacy regulations and guidelines are expected to influence some design decisions especially in terms of the interplay between individual substantive self-control and the legitimate free flow of information. In the following section, various privacy initiatives in the European Union, the US, the Asian Pacific, etc, are briefly discussed.

3.4.1 EU Directive 95/46/EC

The EU Directive 95/46/EC [28] deals with the aspect of protection of Individuals with regard to the processing of Personal Data and on the free movement of such data This directive has two main objectives:

1 "Member States shall protect the fundamental rights and freedoms of natural persons, and in particular their right to privacy with respect to the processing of personal data".

2 Privacy protection should neither restrict nor hamper the free flow of personal data among member states based on the above.

This directive requires that a 'Data Controller' in member states be charged with the responsibility of determining the "purposes and means of processing" of PII "in accordance with national and community laws or regulations" in the context of a "natural person's privacy". In summary, the directive imposes that:

1 The processing of data is done fairly and by lawful means.

2 The collection of data should be carried out for "specified, explicit and legitimate purposes", and not subsequently be processed in a way inconsistent with the stated original purposes. In the case of "historical, statistical or scientific purposes", the processing "shall not be considered as incompatible" provided adequate protection mechanisms are provided by the member state.

3 The collection should be to the "extent that is sufficient and relevant", and must be justifiable and related to the intended purposes.

4 The collection of data "should be accurate and up-to-date". Appropriate safeguards shall be in place to "ensure that inaccurate, or incomplete" data be erased or corrected.

5 The data is "kept in a form which permits identification of data subjects", and "not longer than is necessary", and "only for the purposes for which the data" was originally collected.

6 Data shall not be transferred to third party countries that may not have appropriate levels of privacy protection safeguards.

Furthermore, the directive requires that the data owner be informed adequately in a manner the "purpose for which personal data will be used" is understandable i.e. for some category of data, explicit consent is required before processing of such personal data. Alternatively, the data processing should be necessary for "the fulfilment of a legal obligation" to which the data controller is subject, or be a necessary safeguard for the overall interests of the data owner. An example is where the processing of the data is for the public interest or in pursuit of national security interest, or criminal justice, etc. In this case, the requirement demands that it is the "responsibility" of the data "controller to ensure that the processing" is carried out only for legitimate purposes.

The EU Data Protection Directive for electronic communication (2002/58/EC) attempts to complement the above-mentioned directives. It primarily focuses on aspects of the electronic communications in terms of confidentiality, data traffic and location data. This later directive outlined the restrictions and requirements on the use of Internet browser security features and unsolicited electronic communications in general.

3.4.2 United States Privacy Initiatives

The US privacy activities span public, private and other organizations, covering several specific industry sectors. The US is one region with noticeable privacy activities and concerns due to the various illicit privacy cases that span the public, government, commercial and identity thieves [21, 77, 90]. In fact the sale of personal information seems to be a legitimate commercial activity that has resulted in many abuses of PII in this region [76]. In this light, several sectors have set-out regulations and guidelines including the US Health Insurance Portability and Accountability Act (HIPAA), which deals with how to use and process healthcare related data, the Gramm Leach Bliley Act (GLB Act) [91], concerned with the collection, use and disclosure of PII by the financial service providers. Similar to these acts is the Children's Online Privacy Protection Act (COPPA), which was developed to regulate the collection, processing and use of information about children under 13 years. Overall, these privacy acts focus on notice, and consent, and disclosure of personal information with appropriate security safeguards to ensure data confidentiality and accuracy.

3.4.3 Canadian Privacy Initiatives

The Canadian Privacy Act is a private sector oriented approach to privacy, which specifies how collection, use or disclosure of PII, especially in commercial activities, shall be handled. It covers ten significant areas concerning accountability, consent, limiting collection, identifying purpose, limiting usage, disclosure and retention, accuracy, openness, safeguards, individual access, and the provision of challenging compliance.

3.5 Privacy Protection Information Principles

Independent Organizations in an attempt to broker privacy activities have outlined a set of Fair Information Practices (FIP) regarding the collection, processing and use of PIl. These include, among others, the Organization for Economic Co-operation and Development (OECD) [13], the Online Privacy Alliance (OPA) [26], the Center for Democracy and Technology (CDT) [15], etc. The differences reflect the various sectors' privacy perspectives. The following subsection outlines some of these principles.

3.5.1 OECD Guidelines

OECD is an international organization concerned with global economic cooperation and development. Its guidelines focus on the protection of privacy and the trans-border flow of personal information based on eight main principles [13].

1 Collection Limitation: This deals with the aspect of data collection which stipulates that data be collected by "lawful and fair means", and where suitable with the explicit consent and knowledge of the data owner.

2 Data Quality: Data collection should be appropriate to the purposes for which it is intended to be used, correct, complete and it is up to date.

3 Purpose Specification: This requires that the purposes for which personal information is collected be stated before or at the instance of data collection and further use be limited to the stated original purposes.

4 Use Limitation: The use of personal information shall be compatible with the purposes stated, and should not be disclosed or made available other than to those stated in accordance with purpose specification. Exception is made in the case of a demand with the authority of law or with appropriate consent of the data owner.

5 Security Safeguards: This requires the storage of personal information to be protected by “reasonable security measures against loss, unauthorized access, destruction, modification” and undesirable disclosure.

6 Openness: The use, administration and management of personal information should be transparent as regarding “developments, practices and policies”. In addition, it requires the availability of mechanisms to “establish existence and nature of personal information”, their use purposes and the particulars of the data controller.

7 Individual Participation: This aspect deals with the data owner's participation in the governance of their personal data. In particular, “an individual should have the right to obtain from the data controller, or otherwise, confirmation of whether or not the data controller has data relating to him”, and to have it communicated to him within a reasonable time, perhaps at a charge, if any, but this should not be excessive. Moreover, if the data owner is unable to obtain personal data, when requested, it can challenge any such denial, and if there are misrepresentations in the personal data, they should be deleted or amended accordingly.

8 Accountability: This aspect provides an instrument for allowing the data controller to comply with all measures to ensure the effectiveness and efficiency of the above principles.

3.5.2 OPA Guidelines

The Online Privacy Alliance [26] is an industry alliance of more than 100 global organizations and trade associations which offers guidelines regarding how companies in Internet industry can function in commercial environments in terms of data and information privacy. It describes five main principles:

1. Adoption and Implementation of a Privacy Policy: This mandates organizations to "adopt and implement an appropriate policy framework" tailored to protecting the privacy of individuals' personal data.
2. Notice and Disclosure: This requires that an "organization's privacy policy be made precise, clear, and easy to locate" as well as being available before or at the instance of collection of personal data. It requires that the "policy statement specify the type of information collected", and "whether disclosure to third parties is permissible". It demands that organizations categorically make a statement of commitment regarding data security as well as the necessary steps they aim to take to ensure data quality and access, including addressing the aspects of accountability, disputes and remedy mechanisms.
3. Choice/Consent: The requirement is that the data owner "shall be given the chance to, or options to make a choice" regarding how their PII collected by an organization can be used when subsequent proposed use may be incompatible with the originally stated purposes of collection, including the distribution of such PII to any third party, other than originally stated.
4. Data Security: This aspect requires that "organizations adopt appropriate security measures when creating, maintaining, using or disseminating personal data". In addition, measures should be in place to guarantee "data reliability, protection against loss, damage, misuse, or modification". In the event of lawful disclosure or transfer of data to third parties, it is the responsibility of the organization to ensure that such third parties are aware of the security safeguards on the transferred data.
5. Data Quality and Access: This states the need for organizations to take proper steps to "ensure that the life cycle of information is consistent, accurate, complete and timely" for the mentioned purposes of use. It includes the provision of suitable apparatus for the legitimate rectification of inaccurate data, unreliable data source, collection methods, reasonable access control measures and protection against accidental damage or loss.

3.5.3 The APEC Framework

The Asia-Pacific Economic Cooperation privacy framework [92] is a regional privacy framework for Asia Pacific member communities which regulates privacy activities for the region's mutual benefit. This program is driven by the realization of the economic potential of electronic commerce in today's global business environments, namely cost savings, transparent competitiveness and general improvement of quality of life among citizenry. Similar to other regional initiatives, it seeks to enable regional data transfers with mutual benefit to consumers, businesses, and governments, develop an effective privacy protection mechanism to dismantle potential barriers to information flows, and hence promote economic growth within the APEC region. Most of its principles are derived from the OECD guidelines, except for some subtle interpretations given to some of the components.

3.5.4 Comparative Analyses of Privacy Guidelines and Principles

Central to the philosophy of these laws, legislation and regulations, and FIPs are to devise a set of guidelines and principles aimed at influencing decisions, actions and other activities concerning the collection, processing and sharing of information of a personal nature. The implication is that any piece of information, which explicitly or implicitly can be linked to an identity, should be treated with respect to privacy. It is expected that consumers may have more trust and confidence, if their PII is treated with utmost privacy. This will promote healthier transactions and competition, resulting in greater use of electronic commerce. The balance between legitimate free flow of information and privacy rights can be enshrined by setting the scope, and limits on the collection, processing and use of PII, so that stakeholders can comply appropriately with these regulatory principles that underpin privacy data handling concerns. Having examined a range of privacy directives and principles, it is equally important to put them in the context of this study. It is important to mention that access control infrastructures, which involve the use of PII, should comply with fundamental privacy principles. Given that data collection limitation is widely supported by most of the principles covered, in access control infrastructure, a data consuming party should be made to ask for the minimum set of PII that will satisfy its access requirements. In a typical legitimate access control transaction, the reason for collection is implicit, since the system will require at some point the PII to make an access decision. It also implies that the data owner's consent is implicitly relevant and the intended purpose should be clear. A comparative analysis of these principles with respect to access control infrastructure is therefore imperative and is given below.

1 *Purpose/Notice*: In existing access control systems, usually, *purpose a*nd *notice* are assumed implicit because PII is collected before or at the time of access control operation. However, the above assumption overlooks any subsequent processing or use of the information that may be incompatible with the original purpose of collection. Potentially, to comply with the above principles, the data consuming party should communicate the purpose explicitly in a dynamic way by conveying the policy or requirements to the service requestor, perhaps at runtime. The adverse implication is that this can be vulnerable to probing attack, and business information may be exposed unnecessarily.

2 *Use Limitation*: In access control, the data consuming party should be made to limit the use of PII to the specified purposes (i.e. to determine the access privileges of the data owner in access control operation). Unauthorized disclosure to third parties, subsequent use or disclosure that is incompatible with the original stated purpose should be communicated to the data owner, and if possible, the data owner's consent obtained. This requirement is fundamental, but hard to implement, simply because once PII is in the hand of a third party, controlling its usage is extremely difficult.

3 *Choice/Consent:* Given that choice is the notion of giving the data owner options to say how and when its information should be used and/or processed, consent relates to the secondary use of the data, which in principle should be communicated to the data providing party. Satisfying this requirement in an access control environment would require the dynamic negotiation of service meta-information through perhaps mutual disclosure of access requirements and means of fulfilling the requirements. This in essence gives a party the opportunity to calculate the risks for giving out its resources plus placing obligating constraints on the receiver that should be respected.

4 *Accountability:* This aspect deals with liabilities, enforcement, redress and remedies through civil or criminal enforcement mechanisms. Arguably, in the event of privacy breaches, compromises, disputes, etc, entities usually limit or explicitly refuse to incur liability resulting from representations they make to others in their contracts. In an access control infrastructure, it is important to have a mechanism that would aid in the achievement of accountability. Given the distributed and open nature of the privacy

environment, this would require 'difficult-to repudiate' technical evidence as described in [31].

5 *Data Security and Quality:* This aspect relates to the data quality in terms of reliability, accuracy, completeness as well as the data being up-to-date. In an access control infrastructure, this should be more of a concern to the data consuming party, which relies on the accurateness of the PII provided to make an access decision. Usually, from the consuming party's perspective, it would rely on an underlying trust relationship mechanism to verify and validate the data received from the providing party.

6 *Individual Participation*: This deals with how the individual would be made to actively participate in the handling of information that concerns him or her. It partially, relates to views about user-centric approaches [87, 93] in handling PII. Arguably, individual participation does not necessarily mean that the entity that the data refers to should be the custodian of such data, but it should have a say on how, when, where and who should use or process such data unambiguously. For example, the data owner can allow a bank to transfer some of his personal details to a third party, even though the data owner does not hold the data. Overall, user-centric should imply usage with an appropriate control and consent by the data owner, and can be addressed in context partially by the use of a suitable policy framework, so that an entity can explicitly express its security preferences concerning information about it.

3.6 Conclusion

Privacy has continued to generate concerns due to its growing importance in the current Internet environment. It is a socio-economic problem that has many facets requiring a survey of its problem space. This chapter has presented background material on the concepts of privacy, the notion of anonymity and pseudonymity as privacy safeguards. Furthermore, the chapter has looked into the spectrum of some of the aspects of privacy laws, legislation, FIPs and guidelines. In general, the various privacy viewpoints showed similarities and differences in the perception of privacy by different interest groups. Lastly, the comparative assessment of these guidelines and principles contextually provides insight into how to tackle the privacy problem in the domain of access control.

Chapter 4. Review of the Privacy Characteristics of Access Control Systems

4.1 Introduction

The Internet has forever changed the way some of the enterprise systems are configured due to the need to share their resources with others. In the past, organizations have relied heavily on closed security systems to control access to their resources, and required a form of pre-established knowledge (trusted and known users) between the server and the clients. The closed security systems are no longer robust and cannot scale proportionally in providing secure access due to the need to accommodate users outside the trusted boundary. Thus the benefits and gains of using the Internet for connecting heterogeneous systems are being hampered by the lack of flexibility and sophistication in dealing with increasing security threats [71].

One notable approach in dealing with these threats at application level is access control management systems [4, 6, 94-96], but access control systems make extensive use of identifying information of subjects and objects [6, 22], which raises other security issues and challenges [20, 22, 96, 97]. The emergence of globalization, the increasing pervasiveness of electronic Business to Business (B2B) collaborations, and the upsurge in the use of e-commerce, are growing to a degree that was never anticipated [77]. This, arguably, puts the burden on organizations to provide robust, effective, and scalable security mechanisms to support the restriction of unauthorized access to information assets for both internal and external users. The proliferation of distributed transactions has created the need for enterprises to be able to expose all or part of their applications' functionality to other applications over open networks [3, 98]. Web services help to solve the problem of merging applications that have been developed autonomously and run on a variety of software and hardware platforms.

The Service-Oriented Architecture (SOA) based on Web services is promising to revolutionize the implementation of open and dynamic transactions in many industries [1, 99]. Though web services are more tailored towards application-to-application interactions [1], where access control is a requirement it will make use of some attribute information, credentials, properties or privileges that may identify an entity in order to determine the access privileges of that entity. This raises new security and privacy challenges when services have to be restricted to a certain community of clients. The result is that both service clients and providers are faced with the need to balance security with the increasing demand for prompt access to their information and resources. Access control schemes are an emerging application level security, essentially designed to restrict access to computer resources only to users with provable attribute-identity, properties or privileges. In a ubiquitous environment such as the Internet, users as well as service providers exist in multiple domains with different security preferences and expectations.

This chapter demonstrates extensive investigation of current related work from a technological perspective. It covers the overview of characteristics of open and dynamic systems, with an introductory discussion on authentication, authorization and digital trust. A detailed review is given of a few selected access control management models, especially architectural overviews, with some of their important characteristics. Finally, the comparative analysis of the systems reviewed with a focus on their strengths, weakness and limitations is given.

4.2 Characteristics of Distributed Systems

In distributed environments, there are three functional application level security components that peers in communication have to address: Authentication, Authorization and Trust. Whilst authentication addresses the question of 'who are you?' authorization answers the question 'what can you do?' after we know who you are. In contrast, trust deals with 'who says this about you?', and how the recipient can place confidence in the entity that makes this statement the attesting party. The new federated service architecture (i.e. SOA based on web services) results in a need for authentication and authorization to be managed operationally in autonomous security domains, which requires a trust relationship to be established between those domains. For example, authentication could be handled at the service initiator's domain, whilst authorization could be handled at the service provider's domain based on attributes asserted by a third domain. Consequently trust relationships must be established between all the interacting domains, so that the service provider can trust the authentication statement made by the initiator's domain, and the attribute statements asserted by the third domain. Trust provides the mechanism for validating the authenticity of claims, properties, privileges, etc. that the interacting domains use as the basis for allowing access to protected resources. Trust may be direct or indirect/transitive e.g. A trusts B (direct trust), B trusts C (direct trust) and A trusts C because of B's trust in C (indirect/transitive trust). Often, in distributed environments, the common practice is a reliance on indirect trust brokered by trusted third parties (TTPs) who issue signed tokens or assertions to multiple clients. These clients can then use these tokens, and their mutual trust in the TTP, as a basis for trusting each other. This is how public key infrastructures work today, and it is a foundation upon which federated systems are built.

However, web service provisioning introduces a far more extensive and dynamic environment for complex transactions that makes brokered trust insufficient to protect the privacy and confidentiality of all transactions. Figure 4.1 depicts the multiple actors in federated access management. Service clients can have multiple identities issued by autonomous identity providers, and the identity providers can broker trust among many clients and providers. It is imperative therefore that trust in this case has to be established between people and people, between people and services, and between services and services. This highlights the following challenges and risks:

- What is the accountability of parties in relation to compromised PII?
- What level of audit takes place regarding how PII is accessed?
- How can a user be assured that a service provider's privacy promises will be supported by technological means?

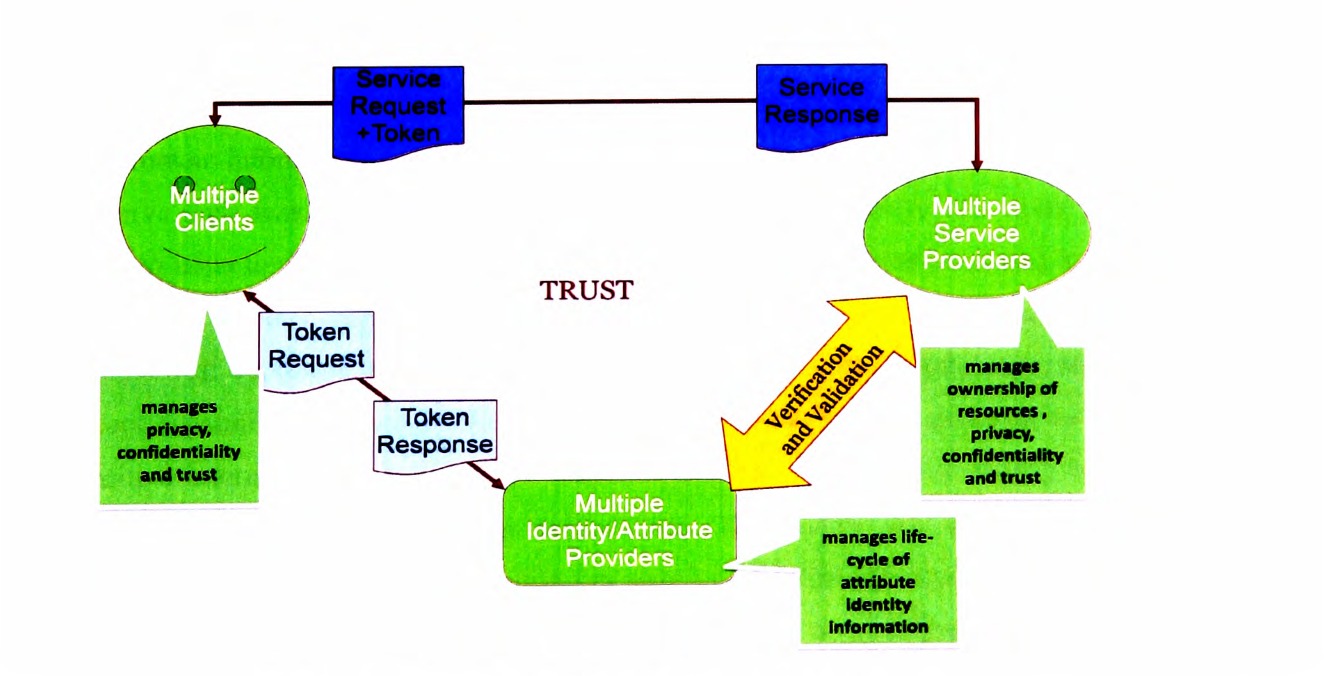

Service Request +Token
Service Response
Multiple Clients
Multiple Service Providers
TRUST
Token Request
Token Response
Verification and Validation
manages privacy, confidentiality and trust
manages ownership of resources, privacy, confidentiality and trust
Multiple Identity/Attribute Providers
manages life-cycle of attribute identity information

Figure 4.1 Actors in Federated Access Management

- How are disputes and liabilities handled? There is a need to establish a respected channel for handling and resolving disputes. How can the process of contractual negotiations be automated? as traditional methods are time consuming and costly.

- Whose fault is it in the event of a problem with shared attributes or data? Who is financially liable? Is there any hard-to-repudiate evidence that parties can use in courts of law to support their claims?

Arguably, the underlying challenges suggest that technological means alone will never be able to answer all the above questions, and that regulatory compliance, disputes resolution and assurance mechanisms will require underpinning with local and trans-border legislation, laws, guidelines and principles. Thus the legal and regulatory system is the only significant trusted third party that is big enough and ubiquitous enough to broker trust between all the parties involved in federated web services. However, technology should be able to contribute to the resolution of the above challenges, by providing high quality, difficult to repudiate information that can be utilized by the legal and regulatory system when the need arises.

Moreover, the emerging web service provisioning may require that in B2B transactions, both parties dynamically exchange service level agreements (SLA) or business level agreements (BLA) in order to assess the mutual benefits and associated risks. This will eliminate the static contractual agreements that are too time consuming to establish, in order to address the opportunities that arise in dynamic business environments. It can be said that BLAs and SLAs may contain information that is of privacy concern, meaning that parties would be interested to preserve their sensitive contractual information. One way to achieve this would be for each party to issue to the other a proof of acceptance of the requirements contained in the SLA or BLA of the other party. Enabling the runtime exchange of these requires a bilateral symmetrical approach, to allow the communicating parties to indicate their willingness to accept constraints imposed by the other party, before the latter is prepared to reveal their sensitive information. It is somewhat debatable whether there is some overlap between user privacy requirements and business requirements. This entails that in distributed systems, remote enforcement of obligating constraints and/or contractual agreements are critical security requirements. Even if controlling the use of digital information at the hand of a remote party is too difficult, these desirable security requirements should be addressed. This is part of the challenges examined by his work and an approach to it is presented in chapter 8. Next, the background of three important security components of a distributed access control is given.

4.2.1 Authentication

In computer network systems, authentication addresses the concept of verifying an identity claim by an entity. In simple systems, this involves verification of a pre-established trust relationship in the form of a username/password pair. In advanced systems, techniques that are more sophisticated may be used, which include multi-factor authentication paradigms, PKI certificates, SAML tokens, Kerberos tickets, etc. Authentication may involve verifying something an entity is, usually biometric identifiers; something an entity has, such as identity cards, cell phones (physical security tokens) or something an entity knows, which includes username–password pair, paraphrase, or personal identification number (PIN). In most cases, using any of these techniques entails that a form of pre-existing knowledge be established between the entity and the verifying systems. In the past, users have had to do this in every application service they attempt to access, but recent developments have advanced authentication to SSO [62, 86]. SSO has emerged as a more flexible way to pass users' authenticated details from one application service to another in an attempt to improve users' experiences. In distributed open systems, this has gone further under the platform of Federated Identity [86] extending SSO across organizational secure boundaries where some kind of federated trust relationships exist among participating members. More importantly, authentication is a vital security service predominately used in network systems to provide the first defense in controlling access to resources, and in most scenarios, may be sufficient to grant access to controlled resources.

At this point, it is useful to define some important terms in the context of this work.

- *Credential:* a set of provable claims used to authenticate the identity of an entity. This includes identifiers for the entity, a proof[3] of the entity's identity, and similar information for the issuer. Where a credential is digitally signed, it may include information such as a digital signature, to indicate that the issuer certifies the claims in the credential.

[3] Whilst it is recognized that absolute proof is not usually possible, the term "proof" here implies that there is some evidence that provides a high degree of confidence in the veracity of the credential.

- *Security token*: a set of provable claims used to authenticate the identity of an entity. This includes identifiers for the entity and a proof of the entity's identity, and for the issuer. Where a token is digitally signed, it may include information such as a digital signature, to indicate that the issuer certifies the claims in the token.

In reality, the difference between a token and a credential is minimal; however, a security token is often associated with the dynamic issuance of short-lived credentials particularly in inter-domain authentication and/or authorization services.

- *Privilege*: a set of allowable access control operations on a target that is assigned to a client i.e. a role, a user, an application, etc; that can make a request to access a resource.

4.2.2 Authorization

Authorization is the process of assigning privileges or credentials; and the determination of whether the capabilities, privileges or credentials of an authenticated entity are sufficient to perform certain actions or functions on a system. In broad terms, authorization is the mechanics of controlling access to sensitive resources based on local rules (access control lists or policies), compared against the access request context provided by the requesting party. The term entity or requesting party may be a human being, a computer system, application, or a network device. In simple terms, the process involves one authenticated party (the service requesting party) submitting some provable property to another entity, the recipient or relying party, that determines the requesting party's access rights and takes a decision. In distributed platforms, attribute based authorization systems are gaining widespread attention but privacy, confidentiality and trust still remain issues that have continued to attract research inputs.

4.2.3 Technical Trust

Trust is another concept that is vaguely understood and lacks any acceptable definition. It means different things to several schools of thought and disciplines, even in the information systems domain. Its use can sometimes be misleading. Our daily living involves the concept of trust which potentially forms part of our decision making process. Though we may not be conscious of our trust instinct, we are still mindful of the degree of confidence we ascribe to people. Similarly, in the technical sphere, trust has a connotation of confidence, though the techniques for determining the degree of confidence may differ. The concern of this study is the technical aspect of trust.

In ITU-T X.509, section 3.3.54 [100], trust is defined as "... an entity can be said to 'trust' a second entity when the first entity makes the assumption that the second entity will behave exactly as the first entity expects". This definition implies that trust has some behavioural properties and/or characteristics, i.e. accepting the proposition of another entity based on the assumption that the entity will act reliably. Even so, in the digital world, it would be difficult to predict a network user's behaviour based on the assumption of an anticipated behaviour. In reality, trust when viewed from an empirical perspective can possess qualitative and subjective factors. That is, trust assumes a providing party's qualitative behaviour, i.e. expectation in respect of the subjective predefined rules by another i.e. the recipient or relying party. Thus, in distributed networked systems, it entails that trust is associated with risks and liabilities and these variables should be balanced to reduce the amount of risk the communicating parties are exposed to at any given time [31]. If the above assumptions suffice, technical confidence relies on pre-established knowledge and the expectation that something unusual will not occur in an unpredictable way. In the digital world, cryptography is one dominant mode of establishing trust, and cryptography is discussed in chapter 5. The subsections that follow discuss some of the concrete access control management models examined.

4.3 Access Control Management Systems

An access control management system lets services control access to resources by requiring a potential client to submit some kind of provable identity information that it can recognize and trust. Several systems have emerged to solve the problem of access to sensitive resources. Some of these systems provide basic access control in the form of authentication, while others require both authentication and authorization. The sections that follow survey some of the popular access control systems.

4.3.1 Kerberos Network Authentication Service

The Kerberos authentication model [101] is a network authentication protocol designed and implemented as a trusted third party service. This service is often referred to as a trusted security arbitrator [102]. The Kerberos authentication framework provides secure authentication that relied on symmetric cryptography initially designed around the Data

Encryption Standard (DES). The primary design goal was to establish an authentication service that relies less on the 'Host Operating System', and aimed at eliminating the reliance on the IP address of the host as the basis to establish trust [103]. To serve a community of users, the Kerberos system shares unique secret keys with every entity (users, network devices, programs, etc) that requests or consumes the authentication service. With the shared unique keys, the Kerberos system can generate messages that can convince parties in communication about each other's identity claims. Kerberos conveys two types of secure credentials called tickets and authenticators [101, 103]. The tickets usually contain the client's name, the network address, the server's name, a timestamp and a session key.

In conveying the ticket, the server encrypts it with the client's unique secret key and can prove the identity for the lifetime of that ticket. Figure 4.2 illustrates a typical Kerberos system architecture and the basic steps necessary for a client to securely converse with a server (service provider) using the Kerberos system. In step 1, the client asks for a Ticket-Granting Ticket (TGT) from the Kerberos authentication service; if this is a human user, (s) he uses a username/password pair to identify themselves to the Kerberos system. In step 2, after the authentication, if the service is satisfied with the identity claim of the requesting client, it generates a session key (TGT) that is encrypted with the client's unique secret key, and then sends it to the client. The client conveys the TGT to the Ticket Granting Service (TGS) and

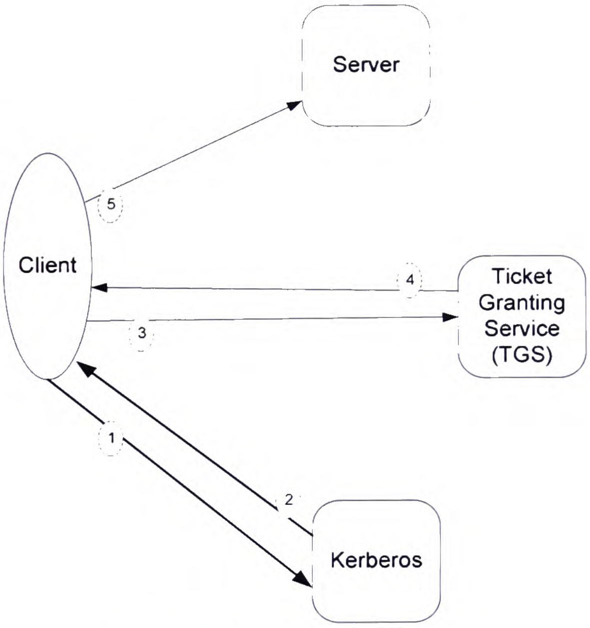


Figure 4.2 The Kerberos Architecture

requests a server ticket in step 3. In step 4, the TGS issues the client a server ticket, which the client can now use to request a service from a server (step 5). The Kerberos authentication service is widely adopted, and is well advanced in integrating with other services to support other security requirements such as authorization. Kerberos has known flaws which include storage of secret keys for every user and server on the network, which must be kept highly secured; otherwise a compromised administrative access would jeopardize the entire infrastructure. It is also possible to cache Kerberos tickets on client systems, giving rise to the potential for an imposter to use an authenticated principal's tickets [103]. In terms of privacy protection, the Kerberos service addresses confidentiality explicitly by the use of shared secret keys among participating entities, but confidentiality does not solve all the privacy problems, particularly when not based on known privacy principles and characteristics.

4.3.2 Central Authentication Service (CAS)

The Central Authentication Service (CAS) is another model ostensibly designed to address authentication across application services based strictly on the HTTP protocol [104]. The primary idea is an attempt to eliminate the need for users to re-authenticate at every service endpoint. CAS denotes a server-client architecture originally developed at Yale University (which became the JA-SIG project in 2004) to provide a trusted central authentication to other applications (clients) in a distributed fashion. Being based on HTTP, each communication endpoint is addressable or referenced by a URI i.e. the login, validation and optional logout points. Its design and implementation is around the Java servlet environment. It's platform independent with a suite of software clients in other programming language including PHP, .Net, ColdFusion, Perl, Java, etc. The clients provide application-level authentication interfaces, which connect to the CAS server. Its trust management capability is based on a multi-tier mode using the idea of signed tickets i.e. opaque strings, to share provable assertions among CASified applications. This is similar to the notion of SSO. Although it does not explicitly handle authorization, it can integrate with other services to provide authorization.

Notably, privacy seems not to be an issue with the CAS system; this assumption may be connected with the belief that authentication using pseudonymity does not reveal PII explicitly. Whilst this assumption is true to a certain extent, in some cases, where authentication is not sufficient to control access to resources, and more attribute-identities are required, privacy becomes an issue. Figure 4.3 illustrates the basic architecture of the CAS service and steps for a user to access a service using the CAS infrastructure. In step 1, the user attempts to access a CASified application service and the CAS client redirects the request to the CAS server in step 2. In step 3, the user authenticates with the server and if successful, the server returns a signed session parameter or ticket, which indicates that authentication, has been successful in step 4. The CAS client verifies and validates the ticket and provides the service to the user in step 5.

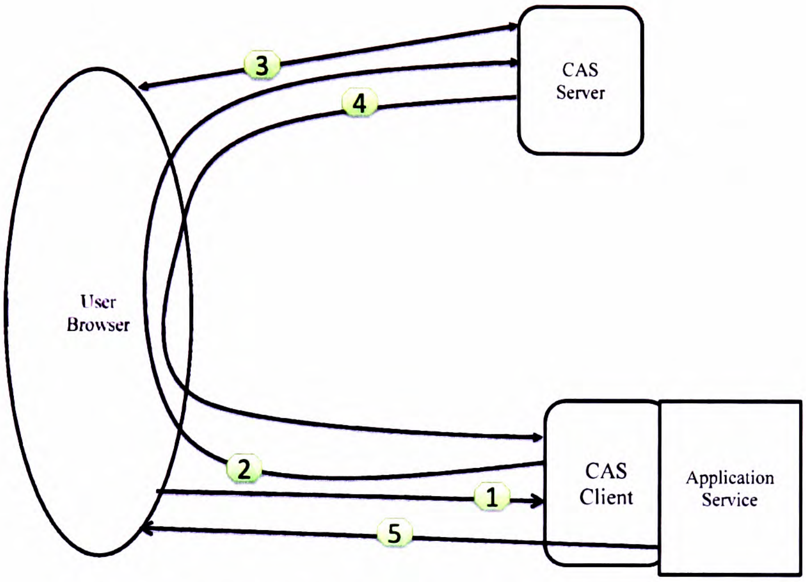


Figure 4.3 CAS Service Architecture

4.3.3 The OpenID

This is a recent decentralized digital identity framework that promises to deliver single digital identity across the Internet, which focuses more on the users' ability to take complete control of managing their identity information. The central idea is a lightweight method of identifying a web user using existing technologies and frameworks [105]. In [105], one acclaimed benefit of OpenID is the ability to provide provable identity without the restriction of having to register or be approved by any third party authority. Similar to other SSO services, it uses existing standard HTTP(S) protocol requests and responses, meaning that it can easily interoperate with the current capabilities of the User-Agent or web based software clients. Another benefit mentioned in [105], is that it is neither tied to the browser cookies or specific identity persistent method of a relying Party nor the OpenID identity provider. However, this identity scheme is still in an infant stage at the time of writing this thesis. Arguably, it will require more layers of security components to address individual user privacy. In [106], it was stated that OpenID based on public domain identifiers, potentially exposes information of a personal nature. Moreover, it is yet to provide a standardized protocol for the exchange of attribute-information that will make it work seamlessly with authorization systems.

4.3.4 Shibboleth Internet2 Middleware

The Shibboleth Internet 2 Middleware [62] is a Federated Identity Management (FIM) system, based on Security Assertion Markup Language (SAML) [107] which was developed for distributed access control management. The primary idea is to allow inter-organizational exchange of user attribute-information to facilitate sharing of protected resources in a secure and trusted manner. Popular in academia, it provides a platform for a secure transfer of attributes of a web-browsing user from the user's origin site - Identity Provider (IdP) to a target site - Service Provider (SP).

In a federated sense, when the user first attempts to gain access to a shibbolized site, they are redirected to a service called Where Are You From (WAYF), which enables them to pick their identity provider (IdP) to perform authentication. This first phase, if successful, prompts the user's IdP service to generate a signed one-time session handle for this user, and passes it on to the target site via the user's browser. The purpose of this handle (a temporary reference) is to enable the target site to ask for more of the user's attributes if necessary, to satisfy other access control requirements before access to protected resources is granted. Furthermore, the handle being opaque is expected to offer the user some kind of privacy protection.

Figure 4.4 depicts the simplified Shibboleth architecture demonstrating typical ldP, SP, WAYE and user interactions. In its basic operation, the user attempts to connect to a resource site and their web browser gets redirected to a discovery service-WAYF to help them pick their IdP in step 1. In step 2, on picking the IdP, the browser gets redirected back to the resource site, which sends an authentication request to the user's IdP. In step 3, the user authenticates to the IdP via a username/password pair. If the authentication succeeds, the user's browser is redirected back to the initial resources they wanted to access. The resource site decides to grant or deny access to the user. In step 4, the resource site may optionally ask for additional attributes of the user to complete the authorization process. The following sections describe the core components of Shibboleth.

Identity Provider (IdP): The IdP is the system that authenticates the user and generates a SAML assertion which is passed on to the service provider. This service is based on the SAML protocols described in SAML core, bindings and profiles specifications. The IdP functional blocks include an Authentication Authority, Attribute Authority, Inter-site Transfer Service, and a SSO as defined in core Shibboleth specifications.

Authentication Authority: This service is based on SAML specifications and provides authentication assertions about the authenticated principals to the relying party i.e. the resource provider. Though Shibboleth in its specification did not specify any method of authentication, it does define the protocol for the exchange of authenticated assertions, based on the Browser/POST and Browser/Artifact profiles.

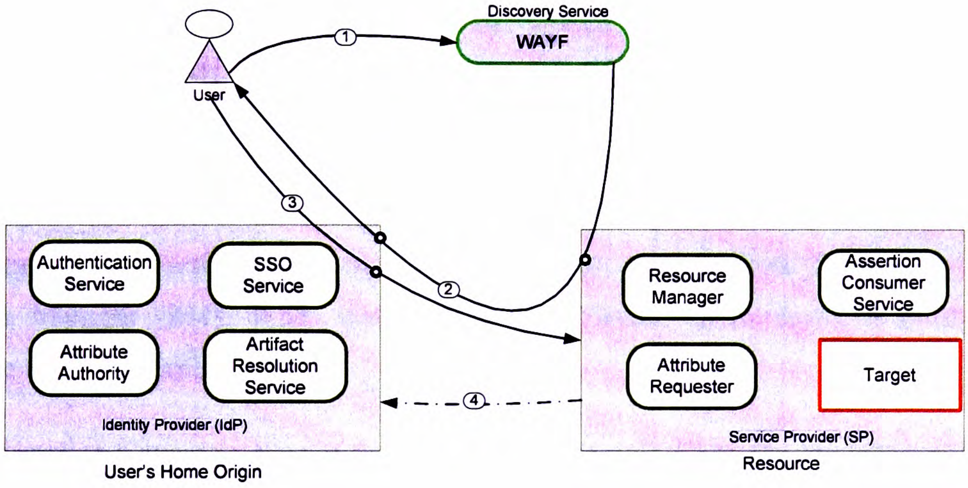


Figure 4.4 Shibboleth Distributed Architecture Framework

Attribute Authority (AA): This service supplies attributes to the relying party based on the SAML protocol binding which comprises <samlp: request> messages containing the <samlp: AttributeQuery> element. This service can perform this function only if the relying party can return an assertion reference (the handle of an authenticated principal). The AA service uses SSL/TLS [RFC 2246] or SAML message signatures to mutually exchange attributes. In Shibboleth, the AA service addresses privacy by using the notion of an Attribute Release Policy (ARP) that governs the release of attributes of principals.

The SSO: The SSO service processes authentication requests from the service provider through the user's HTTP browser capability, and forwards authentication responses to the service provider through the inter-site transfer service.

Inter-Site Transfer Service: This service is based on Browser/POST or Browser/Artifact profiles which interwork with the authentication authority to generate HTTP responses to the user's browser.

Artifact Resolution Service: An artifact resolution service is a system that receives requests directly from the service provider when a Browser/Artifact profile is used and resolves the SAML artifact into the matching assertions.

Service Provider (SP): The SP serves as a gatekeeper and protects services, applications, or resources that are subject to satisfying a set of access control rules. It consists of three primary components: the Assertion Consumer Service (ACS), the Attribute Requester, and the Resource Manager (RM). When deployed in a distributed environment, it must have a reachable unique identifier.

*Assertion Consumer Service (*ACS): The ACS is defined by the browser profile as an HTTP resource controlled service which processes the user request based on the BROWSER/POST or HTTP GET request profiles and must conform to the BROWSER/Artifact profile. Its primary function is to resolve the handle service artifact and establish a new security context for a resource requesting principal.

Attribute Requester (AR)*:* This is the component that is responsible for asking for a requesting principal's attributes, based on the assertion contained in the authentication handle presented by the ACS component. The AR uses this handle in conjunction with the endpoint URL of the corresponding AA to make a request for attributes that it desires, and to which it is allowed, from the AA. The AR uses the attribute acceptance policy (AAP) to perform some sort of validation and analysis.

Resource Manager (RM): This component is the frontline service, which intercepts and responds to requests on shibbolized resources. In normal operation, when the browser request hits the RM, it passes the request to the ACS, and uses the attributes supplied by the AR through the ACS to enforce access.

Where Are You From (WAYF): This is a convenient service that facilitates the discovery of the requesting principal's preferred handle service – the identity provider. This component is usually federated or simply deployed as part of the ACS endpoint to serve as a well-situated proxy to allow users to easily locate and access their identity providers.

4.3.4.1 Privacy Protection in Shibboleth

Shibboleth is the acclaimed first generation of FIM; it recognized the importance of privacy and approached it in two different ways. First, it uses the notion of pseudonyms in the form of opaque strings to shield any identifiable properties of a network user. Second, it uses the concept of *Attribute Release Policy* (ARP) to specify rules that govern access to a user's attribute in case the need arises to reveal it. Shibboleth makes a distinction between site ARP and individual user ARP, which allows site administrators to define default rules for all users. Nevertheless, the implementation of ARP is not in line with any known privacy principles or guidelines, and as a result, it lacks foundational privacy functionality. For example, it has no provision for specifying conditions or obligating constraints on attribute information. More on the limitations of Shibboleth's privacy model will be discussed in section 4.5.

However, Shibboleth is a distributed infrastructure with a tightly coupled trust model that assumes that each Shibboleth entity i.e. origin site, or target site must establish a trust relationship with the others before any exchange of identity attributes. This tight coupling limits its use in a dynamic environment, where trust establishment has to be performed on the fly by previously unknown parties.

4.3.5 The PERMIS Middleware

The PERMIS middleware is a Role Based Access Control (RBAC) model typically based on the ISO 10181-3 Access Control Framework [64], which uses X.509 attribute certificates (ACs) to store Access Control Decision Information (ADI) including policies and user roles. It comprises an access control engine, Privilege Management Infrastructure (PMI) [108] and a GUI policy editing tool [43]. Its core architecture components are an Access Enforcement Function (AEF), and an Access Decision Function (ADF), primarily designed to operate in the same runtime environment [109]. The PMI provides the framework for the management of roles, privileges and role hierarchical relationships using two main administrative components: The Source of Authority (SOA) and subordinate entities each of which is called an Attribute

Authority (AA). The PMI enables the assignment of privileges, delegation, revocation and withdrawal of access rights [110]. In X.509 AC[111], the SOA is typically a service provider responsible for assigning access rights to other entities within a security domain. It supports the delegation of authority with many-to-many relationships, i.e. a SOA can delegate privileges to several other AAs. For example, a SOA (an enterprise security administrator) can delegate AAs (departmental security administrators) to assign privileges to users (within the departments), but the AAs may not be able to certify privileges, although this depends on the delegation policy in operation.

In addition, the PMI architecture consists of two other core components namely: the Privilege Allocator (PA) and Privilege Verificator (PV) subsystems. Whilst the PA is used to assign roles, privileges and access control policies, the PV checks and validates the privileges to determine their trustworthiness. Typically, the PV verifies that a trusted SOA actually issued the privileges, i.e. that the privilege holder has been directly or indirectly authorized by a trusted SOA. The simplified PERMIS architecture is shown in figure 4.5. The AEF and ADF are tightly coupled to operate in the same runtime environment; the idea is to ensure that message exchange between the two takes place in a trusted manner. This makes the exchange fresh and complete so that access requests and responses are presented untampered with [109]. Its access control rules are based on the RBAC specification [63], which can define the "roles" that have what "access privileges", to "which targets" and allowable actions, under what further conditions such access privileges can be allowed. In other words, it assigns rights and privileges based on the roles of potential users of the systems. The following section describes the main policy elements of PERMIS RBAC.

The *<SubjectPolicy>* element describes the characteristics of principals/domains that the rule(s) refers to and is used to determine access to a requested target.
The *<RoleHierarchyPolicy>* is the policy node that defines the various roles and hierarchical relationships (if any) between the roles.

The *<SOAPolicy>* element defines the binding of trust relationships by specifying the SOA/AAs that should be trusted by the system; these trusted SOA/AAs can then allocate roles and privileges to potential subject entities which the system can trust.

The *<RoleAssignmentPolicy>* element describes the roles assigned to subjects/principals and the assigning party i.e. which SOA/AA assigned the roles to the subjects.

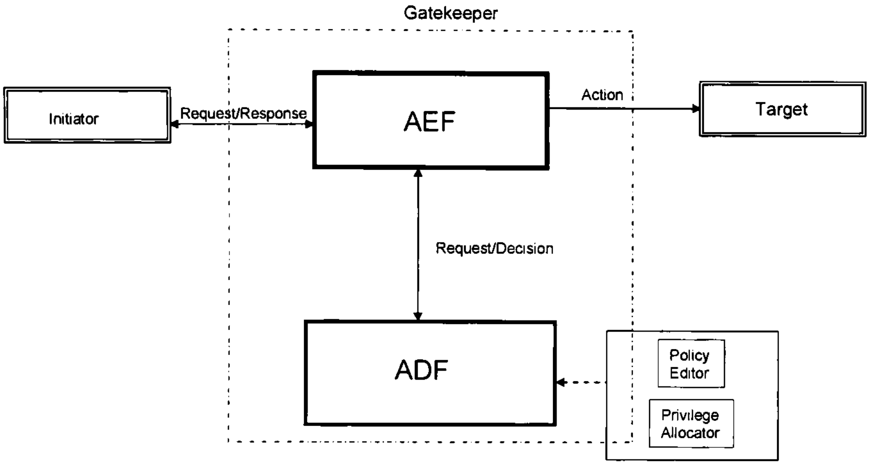


Figure 4.5 PERMIS Access Control Architecture

The *<TargetPolicy>* element defines the target resources covered by the rule policy.

The *<ActionPolicy>* node defines the set of allowable actions or operations supported by the target policy.

The *<TargetAccessPolicy>* node defines the applicable roles, assigns permissions to them, and the actions they can perform on which targets. It optionally includes other environmental constraints that may be imposed by the policy.

In principle, the PERMIS language can be used to control access to PII in terms of confidentiality but not privacy. It is noteworthy that PERMIS has successfully been deployed in many application areas with support for distributed management of trust and delegation of authority as described in [94, 112-114].

4.4 Trust Authorization Management Systems

The concept of the trust management system coined by Blaze et al [115], is another access control approach that has received serious attention within the research community. This has been popularized as a viable paradigm for controlling access to resources [8, 33, 116, 117]. In [32, 118, 119] advances have been made in extending these concepts in distributed authorization platforms where communicating parties may not have pre-established trust relationships. The basic principle is built upon the premise that traditional authorization systems assume an already established trust relationship among communicating parties before runtime. In addition, trust management systems, just as in the real world, assume that a single identity-attribute of an entity [33, 120] is not sufficient to ascribe trust to an entity, and cannot adequately provide the basis for the entity's trustworthiness to access some resources. In practice, more than one digital attribute of that entity would be required to advance the 'threshold' of trust to assign to that entity. These are the underlying concepts upon which trust access management systems are fashioned, in an attempt to improve trustworthiness in open distributed environments.

In trust access management systems, Automatic Trust Negotiation (ATN) has been coined by researchers to refer to the automated gradual and incremental disclosure of policies and digital credentials that can satisfy them. In this case, to advance trust negotiation, resource control disclosure policies are expressed declaratively in a hierarchical manner, which specifies the credentials or properties that a negotiating participant must possess at any stage to satisfy a particular internal subset of a local policy. In bilateral mode, both the client's resources and the service provider's are considered sensitive (services, digital credentials, and policies), and are governed by an access release policy which describes the rules and conditions under which the resource or next policy can be released. This field of study is a well researched area covering trust negotiation concepts [8], models [33, 120], strategies [118], policy languages and digital credentials [121]. In [118] trust negotiation protocols and strategies that describe the sequence and type of messages to exchange are discussed. Similarly in [122], Bertino et al focused on trust expression language and credential formats for describing the resource access conditions and identity attributes respectively. One characterizing component of ATN is the family of negotiating strategies that define the order and sequences of policy and credential disclosures which in most cases is built around the concept of a negotiating tree [8, 118].

In spite of the advances in this area, privacy issues have been identified as one major challenge which has continued to attract research efforts. In [11], Seamons et al recognized two types of privacy vulnerabilities, which they called "probing the possession of sensitive credentials", and "transcript privacy vulnerability" [123]. For example, the release of a driving license may equally reveal other attributes such as age, and address, which a communicating party may not want to disclose. The probing attack happens when an imposter (service provider) can use a bogus service to demand the release of a user's attributes or infer information about another party. In [123], plausible solutions were suggested for how to tackle some of these vulnerabilities using policy migration techniques. Although the suggested solutions can help in resolving inferential attacks, they are not applicable to protecting privacy when resources are disclosed to a remote party. Moreover in [30], Mbanaso et al proposed using XACML policy capabilities to address these limitations and illustrated how negotiating parties can utilize a XACML policy declaratively to control the release of sensitive resources including policies and credentials to mitigate the inferential attacks. They intuitively demonstrated how a policy rule could be used to protect another policy rule, and their evaluations in the context of trust negotiation.

The novelty of the trust access management approach is that the degree of trust can be established incrementally, as more and more policies and credentials are exchanged by negotiating parties. This dynamic negotiation of service parameters, through the gradual disclosure of policies and attribute information, is essential where strangers must build trust before sharing sensitive resources. Figure 4.6 illustrates a simple TN protocol scheme, showing the steps peers in TN can follow to exchange sensitive resources whilst trust is being established. The subsections that follow present some concrete access control systems based on trust concepts.

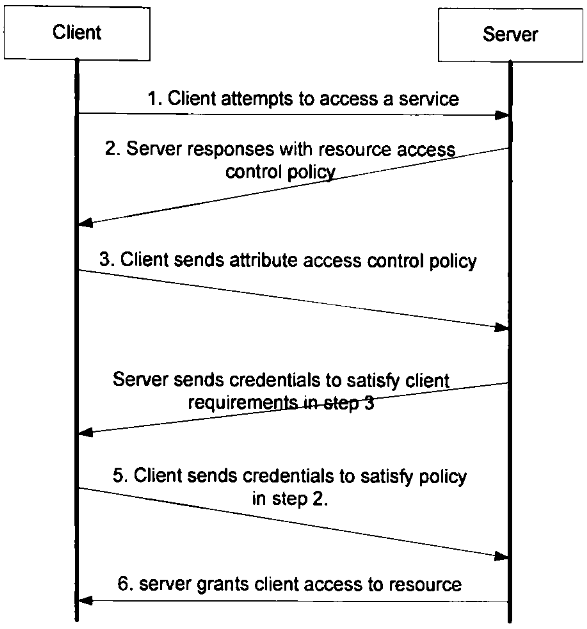

Client
Server
1. Client attempts to access a service
2. Server responses with resource access control policy
3. Client sends attribute access control policy
Server sends credentials to satisfy client requirements in step 3
5. Client sends credentials to satisfy policy in step 2.
6. server grants client access to resource

Figure 4.6 Trust Access Management Basic Protocol

4.4.1 Keynote Trust Management System

The Keynote Trust Management System developed by Blaze, Feigenbaum, Loannidis and Keromytis [124, 125] is popularly believed to be the first trust management system. As often stated by the authors, the design philosophy is a system that treats authentication and authorization using a unified language, so that implemented security policies, trust management components and identity-attributes are uniformly defined. According to [124], KeyNote was designed based on five basic concepts:

- A language for describing 'actions' i.e. the operations are to be controlled by the system plus the consequences of performing such operations, if any.
- A mechanism for identifying 'principals', which are entities that can be authorized to perform actions.
- A language for specifying application 'policies', which govern the actions that a principal is authorized to perform.
- A language for specifying 'credentials', which describes the characteristics of principals allowed to delegate authorization to other principals.

- A 'compliance checker', which provides a service to applications for determining how an action requested by a principal should be handled, given a policy and a set of credentials.

In Keynote, both credentials and policies are collectively referred to as assertions and contain a structured description of rule-actions permitted by the holder of a public key, i.e. credentials and both are expressed using the same syntax. The implication is that its access control process relies entirely on public keys, which must be used by communicating parties to perform access control functions. In [8], Bertino et al drew attention to some of its limitations in handling trust negotiations and stated its lack of compatibility with contemporary TN approaches. Figure 4.7 depicts the KeyNote architecture comprising *Compliance Checker System*, *Credential Management System*, *Trusted Local Policy Source* and protected *Application*. Whilst the credential management system deals with PKI issues, the trusted local policy source provides the policy assertions; and the compliance check makes use of these with request information to issue a response to the application.

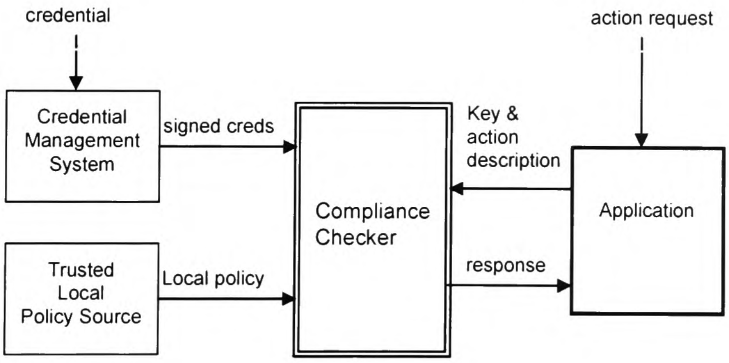


Figure 4.7 Simplified KeyNote Architecture

4.4.2 Trust -X Framework

Bertino et al [126], developed an XML-based framework for trust negotiation, particularly designed for a peer-to-peer scenarios. It consists of an XML encoding language called X-TNL for the representation and formatting of certificates and policies. There are two types of certificates: credentials and declarations. The distinction between the two is that a credential is certified by a certificate authority, whereas declarations are sets of personal information preferences that may not require certification. Additionally, it has the concept of *trust tickets*, a type of certified session identifier that is issued upon successful trust negotiation. The basic idea is to avoid undesirable renegotiation of the same service, which is a performance enhancement tool designed to speed up further interactions. The Trust-X infrastructure is a symmetric architecture like most of the other trust based access control systems. It consists of a *Policy Base* for storing disclosure policies, an X-profile associated with a party, a *Tree Manager* that maintains the state of negotiation, and a *Compliance Checker* that evaluates policies against certificates. Trust-X has two main actors namely: *Controller* and *Requester*, and each actor is characterized by their Trust-X profile of certificates. Whilst the *Controller* is the entity that provides the negotiated resources, the *Requester* is the entity attempting to access the resources. Basically, a negotiating party can act as a *Controller* in one interaction, while acting as a *Requester* in another.

4.4.3 The TrustBuilder Trust Model

The TrustBuilder is a product developed jointly by two universities in the US: University of Illinois at Urbana-Champaign and Brigham Young University. It is based around the basic TN concepts [119]. In particular, TrustBuilder was designed to demonstrate a suite of negotiation strategies described in [118] and presents language-neutral negotiation protocols that must operate within the TrustBuilder architectural framework. It comprises a *credential verification* module, a *policy compliance checker*, and a *negotiation strategy* module as depicted in figure 4.8. According to the authors, the strength of the TrustBuilder architecture is the array of negotiation strategies, which dynamically determine which internal local policy and/or credentials to disclose (and what sequence to follow) based on the current phase of the TN session level already established. Similar to other access management architectures, the credential verification module deals with verification and validation of the received credentials. The policy language and compliance checker uses an IBM Trust Establishment Software engine to perform the matching of the received credential against internal local policy.

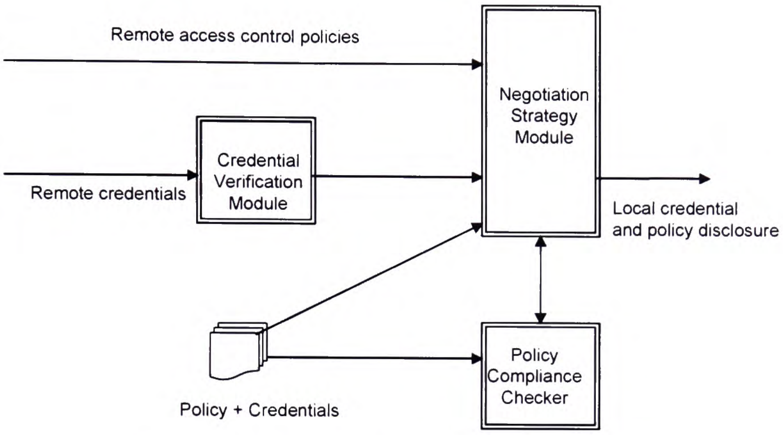

Remote access control policies
Negotiation Strategy Module
Credential Verification Module
Remote credentials
Local credential and policy disclosure
Policy Compliance Checker
Policy + Credentials

Figure 4.8 TrustBuilder Basic Architecture

4.4.4 Browser-Based Trust Negotiation

Recognizing that core TN frameworks have not yet been adapted in standardized Internet security specifications, in [127], Morris demonstrated the use of SAML and XACML in browser-based trust negotiation. He combined SAML 2.0 capabilities, particularly the SSO profile definition and the TN concept, to enable bilateral exchange of information between an AA and SP. He defined a SAML trust negotiation protocol, an extension of SAML request/response, to enable trust establishment between an SP and AA. The SSO protocol adopted here is similar to that of the Shibboleth protocol, but with a bilateral trust negotiation layer, as opposed to Shibboleth unilateral request-response protocol.

In general, although research efforts in trust based access control management models are momentous to open systems security, the unavailability of standardized protocol specifications and common vocabularies will hamper any attempt to independently adapt them into real world applications.

4.5 W3C Web services Architecture Privacy Requirements

The W3C Web Services Architecture (WSA) specified five important privacy requirements that should be considered whenever privacy concerns are security requirements in web service environment. These are outlined below:

- AR020.1 implies that WSA must provide the mechanism enabling privacy policy statements to be described about Web services.

- AR020.2 specifies that advertised Web service privacy policies must be described using P3P.

- AR020.3 the WSA must provide a way a consumer should access a Web service's advertised privacy policy statement.

- AR020.5 the WSA must provide the framework that enables the delegation and propagation of the privacy policy.

- AR020.6: implies that Web Services must not be prohibited from "supporting the interactions where one or more parties of the interaction are anonymous"

These requirements serve as a guideline to applications that use the web services platform; where access control or any form of authentication using PII is involved, it becomes a critical requirement that should be considered.

4.6 Critical Appraisal of the Reviewed Systems

One common denominator of all the access control systems reviewed is that they control access to computer resources using information that can potentially be of privacy concern. Whether the information can be explicitly or implicitly linked to a network user is a matter that relies on the capability of the underlying infrastructure. Access control infrastructure such as Kerberos and CAS specifically provide authentication services that require offline trust establishment. They act as trusted token asserting third parties, though they provide mechanisms for confidentiality but not privacy. The rational for reviewing the authentication services is that the authentication phase is a critical step in protecting privacy in distributed environments. The core access control management systems reviewed share common characteristics including the following:

- A policy language for describing the access control participants' attributes i.e. resource attributes, subject attributes, action attributes;

- Support for specifying credentials of the subject;

- A policy decision point that determines how action(s) requested by subjects on resources are treated.

- A PKI requirement for trust establishment between the various actors.

In the remainder of this section, a critical analysis and assessment of a few of the selected systems is presented.

Fundamental Support for Privacy

The reviewed systems did not take the privacy problem seriously; where privacy is mentioned basic foundational privacy principles are hardly considered in the solution. For instance, the PERMIS infrastructure did not consider privacy from its inception. Arguably, this could be due to its inherent view of the RBAC model, which is chiefly based on the notion of an entity's role. The assumption may be that the role-attribute is unlikely to raise privacy concerns in access control operations. Nevertheless, it has been established in the literature [22, 48] that the role-attribute has the potential to expose a user's capability or profile, which can lead to undesirable privacy risks. In the case of Shibboleth, it is principally designed to allow trusted anonymous authorization using the concept of pseudonymity. In Shibboleth, privacy is said to be addressed in two ways. Firstly, after the user authenticates to the IdP, the Shibboleth authentication service generates a onetime handle to identify the user and transmits this to the SP. Secondly, the IdP uses *Attribute Release Policies (ARP's)* to decide whether to release specific attributes to the SP or not. This is fine as long as the remote site doesn't require any identifiable attributes to complete the service. But this is unlikely to be the case in most transaction scenarios. Although it differentiates between *site ARP* and *user ARP*, which can be combined into the effective *ARP*, the effective *ARP* can simply be classified as a coarse grained access control policy. Besides, the *ARP* is not based on any known standard, and typically does not reflect basic privacy principles or compliance with known FIP, nor does it have support for enforcing obligating constraints. Three main deficiencies of the ARP can be stated as follows:

- The ARP model is inexpressive and structurally defective. It lacks the flexibility and support for expressing fine-grained privacy rules. For example, each identity-attribute is separately given access rights; there is no way to group attributes. It has no environmental expressions for defining environmental conditions and obligations;

- It does not take into account privacy principles i.e. purpose specification, consent/choice, retention period, etc. which are fundamental to privacy safeguards.

The purpose for which the relying party is asking for the release of attributes cannot be determined;

- Shibboleth SPs do not advertise or convey their privacy policy to the IdP; this is vulnerable to exploitation by greedy SPs.

Although Hommel intuitively mapped the ARP into XACML model to extend the ARP capabilities [128], his work never considers how to enforce obligating constraints in autonomous security domains to ensure privacy. In terms of using pseudonyms for privacy, Shibboleth defined standard vocabularies to facilitate the sharing of attribute-information across realms, i.e. origin and target sites, without having to reveal the requesting principal's PII. In a typical transaction, the target site initially identifies a requesting principal by a reference opaque handle. The target site determines access rights based on this handle, but could ask for more attributes of the principal if desirable to complete a transaction. Assuming that the handle is sufficient to allow access to the requested resources, this means that the target site will choose to trust the requesting party based on the trust it has with the IdP. While it can be true that the user's real identity is not exposed in this scenario, an inferential knowledge of the origin of the users may lead to massive profiling over a time. Moreover, the undesirable disclosure of a *role attribute* can reveal the *capability of a principal*, which in some environments may constitute a privacy risk. The fact that the principal is initially known to the target site by a onetime randomly generated handle, and subsequently associates with role attribute(s) raises a privacy and confidentiality concerns.

Arguably, privacy in this sense using pseudonyms is vulnerable since there are no strong privacy obligations and bindings between the principal and the target site. Simply put, trust in this case is typically transitive, which cannot give privacy assurance in high value transactions. Furthermore, the use of an opaque handle, which shields the real identity of a principal from the target site, is fine as long as the target site does not require any more identifying attribute-information of that principal to complete a service request. Nevertheless, this assumption is unlikely to be the case in many dynamic transactions where more user attribute-information may be required in order to complete a transaction. Consider this classic example of a University of Salford student who wants to access the University of Manchester's web services. The student authenticates through his IdP (University of Salford) to obtain a token before invoking a service. The token attests the fact that he is a bona-fide student of the

University of Salford.

However, during the session, the student needs to access nuclear related material that requires an attribute-identity type: *nuclear scientist*, before the service can be invoked. The student may have some privacy concerns for this attribute. As the student has no way to express and enforce his privacy rules at this target site, he faces a privacy risk. In other words, the fact that there are no obligating constraints on the side of the service provider potentially places the student's privacy in danger. In the case of the TrustBuilder architecture, privacy is not plainly considered; fundamental principles for safeguarding privacy are overlooked. However, it did consider vulnerabilities such as probing and inferential attacks with respect to privacy. It is disputable whether the approach is more about confidentiality safeguards than privacy.

Trust Establishment

All the systems reviewed required one form of trust establishment. The client is required to use a form of provable identity attribute issued by a mutually trusted TTP in order to invoke services. The provable identity may be in the form of a username/password pair, a certificate or token. In a distributed sense, the public key certificate is often relied upon to provide the trust mechanism across boundaries of trust, and it is the foundation upon which federated systems are built. Most of the reviewed systems rely on the public key certificates for establishing trust between security realms.

However, the appraisal of privacy principles in chapter 3 exposes the fact that trust purely based on PKI is insufficient to guarantee privacy protection in ubiquitous environments, since third party PKI issuers can hardly vouch for how the certificate holder handles other parties' information. Automatic trust negotiation is one approach that attempts to enable bilateral building of trust using other properties of communicating parties to improve trusted sharing of information.

However, PERMIS and Shibboleth systems are typically unilateral in the sense that they do not provide a mechanism to negotiate between the service client and provider. The client cannot ascertain the trustworthiness of the service provider before giving out its attribute-information. The unilateral paradigm reflects one way trust assessment; the service provider can verify and validate the claims of the client without the client being able to do the same. The inability of the client to negotiate for the release of its attribute information potentially endangers its privacy. In contrast, trust management systems favour bilateral and incremental negotiation, using access control policies and credentials. Trust management systems are traditionally designed to allow previously unknown parties to build trust and exchange sensitive resources. The contrary is the case with the PERMIS infrastructure, which requires some form of pre-registration of users by offline means. In case of Shibboleth users are only required to establish offline trust with their origin sites. In many distributed transactions [8, 117] pre-registration seems impractical and uneconomical for all users who are not within the same security domain.

Moreover, in situations where users are unaware of the access requirements, the likelihood is that more identity information than required may be disclosed by the client, which defeats *minimal disclosure* principles. One approach is for SPs to advertise their sets of access requirement policies to allow clients to determine which attributes to supply in a given service invocation. But from the service provider's perspective, advertisement of complete access requirements may expose some business information, the undesirable release of which to arbitrary strangers can put such business at risk. The consequence is that the absence of an effective privacy negotiation mechanism could mean that a party may simply be subjected to giving more credentials than are necessary.

Privacy Disputes and Liabilities

None of the systems reviewed mentioned compliance with legal aspects of privacy. It is apparent that identity-information collected by service providers is most often stored in the corporate repository, to which the organization purportedly controls access on behalf of the owners. The implication is that a user may not have the option to choose who to entrust his PII to, and may have to cope with the promise that the information will be accorded privacy respects and protection. Moreover, in most of these cases, the organizational data handling policies are static and apply to everyone whose personal information is held by that enterprise. The enforcement of a single static policy on a huge amount of personal data can rarely reflect the privacy preferences of each owner.

The above inferences lead to the conclusion that technological means alone will never be able to answer all the above questions, and that regulatory compliance, disputes resolution and assurance mechanisms will require underpinning with local and trans-border legislation, laws, guidelines and principles. Thus the legal and regulatory system is the only significant trusted third party that is big enough and ubiquitous enough to broker trust between all the parties involved in federated services. However, technology should be able to contribute to the resolution of the above challenges, by providing high quality, difficult to repudiate information that can be utilized by the legal and regulatory system when the need arises.

It is important to note that the TN systems contributed immensely in shaping the search for alternative solutions, especially in the consideration of privacy assurance and remote enforcement of obligating constraints. Overall, this study reveals that the traditional assumptions that privacy concerns are on the service client's side no longer holds. It has been established in the thesis that both the server and client sides have privacy and confidentiality concerns that require simultaneous protection. Moreover, new business transactions may impose dynamic composition of services on one hand, and the exchange of service requirements, agreements, constraints and credentials on the other, making dynamic negotiation a novel approach.

4.7 Conclusion

The chapter has provided the basic elements of access control systems in the context of privacy, confidentiality and trust in distributed environments, and has further examined privacy from the perspective of concrete access control implementations. The literature review has established the distinction between preserving the confidentiality of service outputs and the privacy of service meta information i.e. identities, attributes, policies etc. The dissimilarities in both types of resources have become blurred, and preserving them seems similar and symmetric. The fact that communicating parties, be it the service client or provider, are likely to possess information that requires some degree of confidentiality and privacy protection gave new insight into the kind of privacy infrastructure to develop. The potential for a digital resource to be persistent (i.e. having a property of reusability) supports the necessity for a privacy solution that is capable of ensuring privacy and confidentiality across multiple security domains.

The critical analysis and assessment of similar work exposed the need to alter the current approach to developing privacy aware systems, which favours support for symmetrical architecture. This complements the need for dynamic negotiation, which can allow communicating parties to indicate their willingness to accept constraints being imposed on their use of information provided by either party, before sensitive resources are exchanged. The idea is to treat any communicating party's sensitive information with a concern for their privacy and security preferences. Moreover, since parties' degree of trust threshold may change based on perceived risks or knowledge due to business and/or security expectations, the dynamic exchange of security requirements can offer both service and client the opportunity to weigh service benefits against security risks. To sum up, the analysis and assessment of the privacy characteristics of these selected systems open up new knowledge that should influence decisions in designing a privacy aware access control systems.

Chapter 5. Privacy Enhancement Technologies and Security Standards

5.1 Introduction

Historically, making information unintelligible sprang from the requirement to secure military and political conversations. Consequently, codes and ciphers were devised to obscure messages, and make them unintelligible or unreadable to any interceptor or unintended recipient [129]. Before the dawn of telegraphy, telephone and radio communications, obscuring information could be traced as far back as 6BC - the 'scytale' used by the Greeks to cipher messages [129]. Also, during the time of Caesar, codes were formulated to mask information from strangers, so that adversaries were kept away from information classified sensitive and strategic. Most of these primitive ciphers were permutations and substitutions of characters. In Caesar's cipher, a simple substitution mechanism of alphabets based on the manipulation of the position of letters in an ordered fashion, (i.e. by the interposition of the alphabets in an understood manner) was devised [130]. In this scheme, the letter *A* can be replaced with the letter *M*, and *B* is replaced with *N*, etc and this pattern was preserved to revise the information to its original format [102].

The backdrop of making information unreadable is a result of the human 'survival instinct', characterized by lack of trust and the quest for domination. The need to keep some secrecy and appear discreet has continued to drive making information unintelligible, chiefly done for personal, political, warfare and business purposes. This is in fact part of the human survival instinct, which tends to achieve competitive advantage over a party or competitors by attempting to gain knowledge of the other party's intentions and capabilities., Although these techniques may simply appear to be insecure by today's standards, monarchs and military dictators used them effectively to secure their conversation from unintended audience. The advent of computers and communication networks changed the landscape of information security. Initially, the battle was how to secure the computing resources against disgruntled staff, in order to stop them causing harm to the underlying infrastructure. The evolution of computer networks spanning the public domain brought about the Internet which further placed enormous challenges on information security.

The interconnection of computing devices, their subsequent central control and the fact that there are a large number of users raised another important dimension in securing information

[130]. The drive to enshrine trust and accountability in the use of computing systems then emerged. Apart from making information unreadable to an unintended audience, the need arises to limit what typical users could do with interconnected systems. This ushered in the notion of limited user accounts and unlimited administrative accounts to secure the backbone infrastructure from potential misuse by using simple usernames and passwords [130].

Thus computer security still revolves around the use of codes and ciphers to manipulate the underlying system in such a way that it becomes no longer useful to unintended entities. However, modern computer security has advanced through the evolution of cryptography [102, 129]. The earlier systems include wheel Ciphers (mechanical devices used by the Germans in the Second World War, and passphrases used as keys to protect data (Giovan Belaso, 1500s). These encryption methods are popularly known as symmetric ciphers. At the turn of the 1970s effort to standardized cryptography mechanisms brought about the Data Encryption Standard (DES), which uses a 56-bit key to encrypt and decrypt data [102]. Subsequently CAST, RC2 and Triple DES were developed to overcome the weakness of earlier symmetric systems.

This chapter examines the general concepts, principles and techniques that have evolved over the years in an attempt to protect computer and networks against threats, and highlights some frameworks that have continued in the advancement of their developments. In particular, this chapter overviews the development of cryptography, and how it has contributed immensely to the development of privacy enhancement tools. Additionally, it introduces some Internet standards and specifications including Secure Socket Layer (SSL), and Transport Layer Security (TLS). (Chapter 6 will look more closely at the specific standards that are used in this work.)

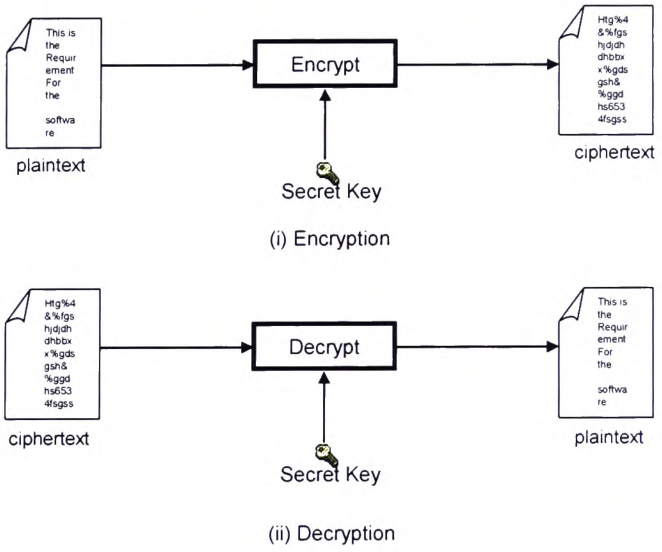
5.2 Cryptography

Inherently, computer security mirrors the normal physical security that we practice in our daily lives. We lock our house to prevent unwanted visitors from having access to it and secretly safeguard the keys. Similarly, in the computer world, to keep resources away from strange parties we secure them using some sort of lock and safeguard the keys. Cryptography, which mimics the physical locks and keys, emerged to help secure the digital world of zeros and ones.

Cryptography is a field of mathematics and computer science, typically concerned with information security in what mostly happens in two modes: encryption and decryption [102, 130], and has evolved through empirical and theoretical design efforts. The underlying critical factor is the concept of a secret key, which is fundamental to its security. Several studies have argued that as long as the key remains undisclosed to an unintended audience, assuming the unavailability of powerful computing resources to mount an attack, the security will remain uncompromised [102, 130]. It evolved from the need to provide stronger methods of securing computer networks in terms of authentication, authorization, integrity, etc, which are fundamental to privacy and confidentiality protection. Cryptography has two main models: symmetric and asymmetric. These models in many ways have characterized the development of privacy and security enhancement technologies.

5.2.1 Symmetric Cryptography

Symmetric cryptography implies using the same secret key for transforming the information into gibberish (ciphertext) and for converting the gibberish back to the original text. In this, as mentioned previously, the secret key is fundamental to the security it provides. A variety of symmetric cryptosystems (algorithms) have been developed over the years including shift ciphers such as Caesar, ROT-1, modern symmetric ciphers such as DES, IDEA, RC5, CAST-128, AES [102, 130] , etc. In modern cryptography, computing algorithms, and mathematical logic are used to manipulate the processes that transform information into unreadable formats and the reversing of the same into readable forms applying the same secret key. Figure 5.1 depicts the process of using the same cryptographic key to encrypt a piece of data and decrypt


This is the Requir ement For the softwa re
plaintext
Encrypt
Secret Key
Htg%4 &%fgs hjdjdh dhbbx x%gds gsh& %ggd hs653 4fsgss
ciphertext
Htg%4 &%fgs hjdjdh dhbbx x%gds gsh& %ggd hs653 4fsgss
ciphertext
Decrypt
Secret Key
This is the Requir ement For the softwa re
plaintext

(i) Encryption

(ii) Decryption

Figure 5.1 Symmetric Cryptography

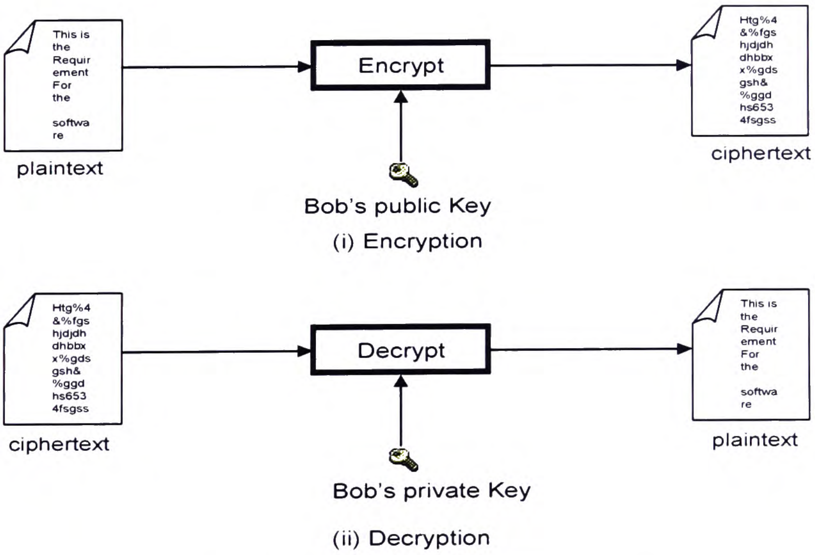
it back to its original form. For example, if Alice wants to send a secret message to Bob, Alice must share the same key with Bob in advance before any ciphertext can be transmitted. In addition, this requires the sharing of the key to be carried out separately out-of-band and in a secure manner. This out-of-band sharing of the key brings an additional communication overhead in the distribution and management of the keys. This is the main drawback of this model. More so, if Bob wants to send message to a community of five hundred (500) people in a confidential manner, Bob potentially needs 500 secret keys. For a community of *n* users, the number of unique secret keys can grow up to ½(n^2 - n). Consequently, there is an overhead in the manageability of the key data in terms of maintenance, storage, etc.[131]. For example, if Alice, Bob, Charlie and Dan want to send several distinctive messages to one another, they must each generate and share a unique secret key with each other; otherwise the confidentiality of their conversation could be compromised. This case is worsened, for instance, if each person has to generate a fresh unique key to encrypt the messages for every recipient each time [102]. Furthermore, the requirement to share keys out-of-band prior to confidential communication makes symmetric cryptography difficult to use between parties without pre-existing trust establishment. If Alice wants to send PII to Bob, who she has never had any previous conversation with, she cannot be certain that she is sharing her key with Bob and not Dan, an impersonator who is posing as Bob in order to collect her PII.

5.2.2 Asymmetric Cryptography

The key management and distribution problems inherent in symmetric cryptography prompted extensive research efforts. In the mid-1970s, two notable researchers: Whitefield Diffie and Martin Hellman presented to the public a major breakthrough that shaped new directions in cryptography [132]. They postulated the possibility of using a key pair in performing cryptographic functions. The postulate is such that one key can be used to perform encryption and the other for decryption. The novelty was generally accepted to have fuelled modern inventions in cryptography, popularly known as public key cryptography [129, 131, 133]. Public key cryptography, also called asymmetric ciphering, no doubt changed the landscape of information security. The idea of asymmetric cryptography is a process whereby a pair of mathematically related keys is employed in transforming data into gibberish and transforming back to the original data. This gives the notion of public and private keys. The idea is that one of the keys can be made public (stored in directory, database or listed in a public telephone book, etc.) without the risk of compromising security as long as the other key remains private [102, 131].

It is often stated in the literature that, despite the fact that the key pair is mathematically related, knowing one key is not sufficient both mathematically and computationally to derive the other. The dependability of an asymmetric cryptosystem is based on the belief that it is computationally infeasible to derive one key from the other, without the disclosure of the underlying mathematical factors. Figure 5.2 illustrates the basic concept of an asymmetric system, showing how Bob's key pair can be used in the process of encryption and decryption. For Alice to communicate with Bob securely, she uses Bob's public key (P_{UK}), to encrypt the message and Bob can use his private key, only known to him, to decrypt the message.

Notwithstanding this extraordinary advance, it has generally been accepted to have some inherent drawbacks [102]. One notable flaw is that asymmetric ciphers are quite computing intensive and therefore slow in process. This drawback often limits it to ciphering small chunks of data, whereas symmetric systems can perform cryptographic operations faster in bulk data operations. Although advances have been made in asymmetric cryptography in recent years, through algorithm developments such as elliptic curve algorithm, this remains an issue.


This is the Requir ement For the softwa re
plaintext
Encrypt
Bob's public Key
Htg%4 &%fgs hjdjdh dhbbx x%gds gsh& %ggd hs653 4fsgss
ciphertext
Htg%4 &%fgs hjdjdh dhbbx x%gds gsh& %ggd hs653 4fsgss
ciphertext
Decrypt
Bob's private Key
This is the Requir ement For the softwa re
plaintext

(i) Encryption

(ii) Decryption

Figure 5.2 Asymmetric Cryptography

However, many modern security mechanisms combine the capabilities of these two models to provide highly secure environments in ways that are not possible using any of the models alone. To benefit from the key management and distribution capabilities of the public key system, the notion of public key infrastructure (PKI) [131] was developed to facilitate the sharing of P_{UK} in a tamper-resistant manner[4]. This development has changed the landscape of security in distributed network environments, allowing previously unknown strangers to communicate in a secure manner. The public key cryptography provides the platform for enabling important security services, including the digital signature scheme discussed next.

5.3 Digital Signature Scheme

In the physical world, a handwritten signature on a document binds the signer to the content, and it can be verified and validated by other relying parties. Similarly, in the digital world, a digital signature provides a means of signing documents digitally. A digital signature can be verified and validated, thus confirming the identity of the entity (individual, organization or system) that made the digital signature. The digital signing process in an asymmetric cryptography context is a private key operation on data, which results in a signature value, and a public key operation on the same data to verify the signature and check whether the value corresponds to the original value [102, 130]. Usually, a cryptographic *Hash* function [102, 134] is used to transform the input data (of any size) into a fixed-size output called the digest. Figure 5.3 depicts a public key digital signature scheme. To sign data, the signer performs the operation in two steps:

- It transforms the data into a fixed-size value (digest) using a *Hash* function,
- It encrypts the digest with its private key, which results in a signature value.

In order for the relying party to verify and validate the signature, the original message must be conveyed with the resultant signature, so that the recipient can perform a reverse operation. Similarly, the relying party takes the same two steps in verifying the signature.

- It hashes the received data to a fixed-size value (digest) using the same *Hash* function,

[4] Although a certificate is used to guarantee the public key(s) is that of the person who claims to own it, there is a need to ensure that an entity's certificate is not tampered with.

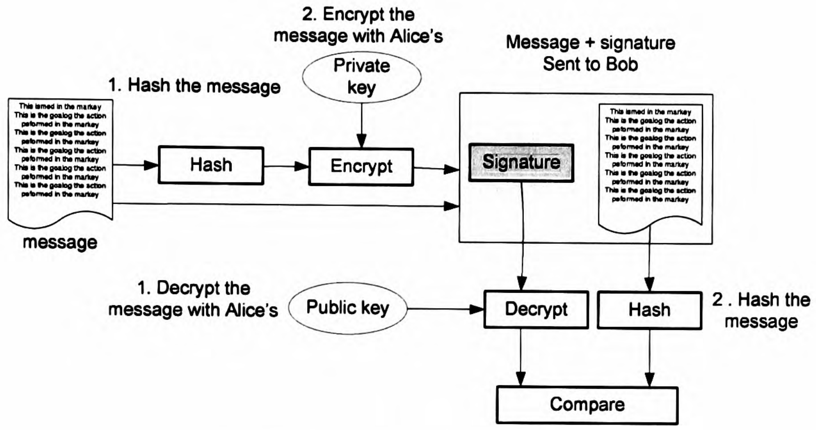


Figure 5.3 Digital Signature Scheme

- Using the sender's public key, it decrypts the signature, and then compares the result against the computed digest. If the computed digest matches the decrypted digest, the signature is verified.

In other words, the successful decryption of the digest proves the fact that the sender actually signed the document (assuming the signing key remains secret and can be proved to belong to the holder) and the successful comparisons of the digest with the digest generated by the recipient attests that the original message has arrived as sent. This scheme addresses two important security services: non-repudiation - which thwarts Bob from going back on his digital claims, and data integrity- which provides the property of unforgeability, i.e. Charlie cannot alter the message Alice sent to Bob. Overall, the idea of using public key cryptography in digital signature is that it is computationally infeasible (expensive) for someone to forge Alice's signature, and Alice cannot deny (non-repudiation) that she signed the document[5] as long as her private key remains private.

The *Hash* functions are based on one-way cryptographic algorithms that transform an input data of arbitrary size to a fixed-size output (e.g. 160 bits for SHA-1, 128 bits for MD5). The

[5] Though Alice can still claim that she did not sign the document by asserting that her key was lost, other security measures are in place to ensure that Alice cannot deny the fact that she signed the document on the basis of compromise.

assumption is that the one-way function makes finding two different hash inputs that give the same hash output computationally expensive. Notwithstanding this, there has been a number of attacks on digital signature schemes, which can be found in [134, 135]. Over the years, a variety of *Hash* algorithms have emerged including the Digital Signature Algorithm (DSA) described in NIST- the U.S. Department of Commerce publication [136], the MD family [RFC 1319, RFC1320] and particularly, MD5 [RFC1321] developed by Ron Rivest, the Secure Hash Algorithm (SHA) family [SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512] also published in NIST publications [137]. In general, the strength of the digital signature to a large extent depends on the *Hash* algorithm used in generating the digest, and its length. Notably, short length digests are potentially vulnerable to attacks [102, 135] such as birthday attacks. In practice, the recommendation is to use *Hash* functions that can produce outputs that are long enough to make the birthday attacks potentially infeasible.

In summary, digital signature schemes provide important security services such as proof of authenticity, integrity and non-repudiation. Consequently, the relevance of these services is crucial, especially in the context of guaranteeing the remote enforcement of privacy and confidentiality as described in chapter 8. In practice, there are limitations in the use of digital signature which will be discussed further in chapter 10.

5.4 Public Key Infrastructure (PKI)

The full benefits of Public Key Technology (PKT) could not have been realized without devising a trustworthy way to address the key distribution and management problems raised by the use of cryptography in large-scale public networks. Computer security is often based on the concepts of trust, knowledge and to some extent accountability. The PKI provides such an enabling framework that provides support for the binding of identity (and other identity attributes) to a public key through the process of registration and issuance. Securing disparate systems requires this pervasive infrastructure, which is not only available to the participating community, but can also be scalable proportionally in many respects to provide the needed security across multiple domains.

Moreover, there is a need to ensure that the public key (or other attributes associated with the key) is not tampered with, that is, the binding of the public key to a claimed holder must be done in a trustworthy manner. It implies that the relying party has to be assured of the authenticity of the public key. By using the PKI, the identity attributes, public key, the binding operation and validity are made unforgeable. This makes it possible for entities without previous contact to authenticate each other and exchange data in secure fashion. Today, most essential security services used in open networks to provide trust, authenticity, integrity and confidentiality are enabled, to a large extent, by PKI services. Outlined below are some of the components of PKI, and the highlights of the services they provide:

Certification Authority (CA): In PKI terms, certification can be described as an act of binding an identity[6] with public key. Accordingly, the CA can be described as a trusted third party that certifies and digitally signs a data structure containing some representation of one or more attributes of an identity and a corresponding public key [131]. The concept of a CA provides a powerful channel for the issuance and distribution of public keys in large scale environments. It enables entities to make their public P_{UK} component available and accessible to a communicating community. A holder of a P_{UK} must convince others that the P_{UK} component actually belongs to it, and has not been altered, i.e. proving ownership and integrity of the key. The *Public Key Certificate* (PKC) was invented to bind an identity to public key(s) which must be digitally signed by one or more CAs. One variant of the PKC is specified by the X.509 ITU-T standard [138]. The X.509 ITU-T standard specifies standard formats for public key certificates, certificate revocation lists, a certification path validation algorithms, etc. The generic ITU-T X.509 certificate structure is described in section 5.3.1.1.

Certificate Repository: Whilst the concepts of an X.509 certificate and CA are convenient mechanisms for making P_{UK} available, the absence of an accessible repository or directory to store and easily locate the PKC will defeat its gains. Although in a small user community, it is possible for an entity to disseminate its PKC using out-of-band mechanisms, in a large-scale network, manual distribution of the PKC causes both administrative and security bottlenecks [131]. For example, a private key can be compromised; dealing with situations where keys are compromised will not only be too difficult to handle by individuals, but the exchange of revocation may also be unreliable. Thus, the need to manage the life cycle of P_{UK} is critical to the overall security of PKI. The provision of an accessible location for the PKC solves the problem of key distribution, and provides a means to check for validity. The certificate

[6] It is possible to include other attributes of the identity.

repository is a logical central storage system capable of handling the life-cycle management of the PKC and its revocations. The RFC2538 [139] specifies the Domain Name System (DNS) for storing certificates including LDAP servers, X.500 Directory System Agents (DSAs), web servers using http and ftp (RFC2585) protocols, and corporate domain database systems.

Certificate Revocation: As previously mentioned, a private key can be lost or compromised which invalidates the use of the corresponding certificate. In practice, the relationship between a CA and a key holder may change, thus breaking the binding and its validity. The CA needs a convenient mechanism to alert the community of users that such a certificate can no longer be trusted, or that it has ceased to vouch for the certificate. This component enables the CA to publish its certificate revocation list within the same repository making it possible to verify the expiry of certificates before their use.

Other services of the PKI include *Automatic Key Update*, *Cross-Certification*, *Key Backup and Recovery*, *Key History*, etc. These services make up a comprehensive PKI and are necessary to realize the full benefits of public key cryptography. Readers interested in PKI can find more detailed information in [131].

5.4.1.1 The Public Key Certificate

Public key cryptography makes use of a key pair: public and private keys to perform cryptographic operations. The attractiveness of public key cryptography from the perspective of scalability is the fact that the public key component of the key pair could be distributed freely among the community of users. To support a wider scale of services provided by the public key technology, it is needful to assure the public key relying party (i.e. the entity that uses the public key and associated attributes for some security services) that:

- The integrity of the public key (and other associated attributes) is not tampered with.
- The binding of the public key (and other associated attributes) to the claimed holder has been done in trustworthy manner.

The concept of public key certificates was invented to fulfill the above stated goals [131]. Thus a public key certificate can be described as a data structure issued and digitally signed by a trusted third party (the issuer and signer), which binds an identity (and other associated attributes of that identity) with the corresponding public key of the claimed holder. For the sake of interoperability, a variety of standards that define the structure and semantics of public key certificate, including X.509 [138, 140], Simply Public Key Infrastructure (SPKI) [141, 142], OpenPGP [143], etc, have emerged in recent times. Figure 5.4 shows the generic structure of X.509 certificate version 3 adapted from [131] [144] showing its fields which are described below.

- The *Version* describes the certificate version number (e.g. 1, 2, or 3).
- The *Serial Number* indicates a unique identifier that identifies the certificate within the issuer's domain.
- The *Signature* describes the identifier of the digital signature processing algorithm used (i.e. the Object Identifier (OID)[7] and other parameters) e.g. an Object identifier for SHA-1 with RSA indicates that the hash function algorithm is SHA-1 encrypted using RSA public key algorithm.
- The *Issuer* describes the CA that issued and signed the certificate, and is usually indicated by the Distinguished Name (DN) of that CA. This field must be non-null.
- The *Validity* indicates the start and end dates the certificate should be considered valid.

[7] OID is a hierarchically globally interpretable identifier to identify mechanisms and name formats; the sstructure and encoding of OID is defined in ISOIEC-8824 and ISOIEC-8825.

- The *Subject* specifies the name or DN of the certificate holder, and must be non-null.

- The *Subject Public Key Info* specifies the public key and the associated public key algorithm used in generating the holder's key pair.

- The *Issuer Unique ID* specifies a unique identifier of the issuer, and could be present in versions 2 and 3 only[8].

- The *Subject Unique ID* specifies a unique identifier of the certificate holder, and could be present in versions 2 and 3 only[9].

- The *Extensions* is an optional field present in version 3 only, and is used to specify recognized options and private extensions. Details of these standard options are detailed in [131].

The *Digital Signature* contains the resultant signature block and algorithm identifiers.

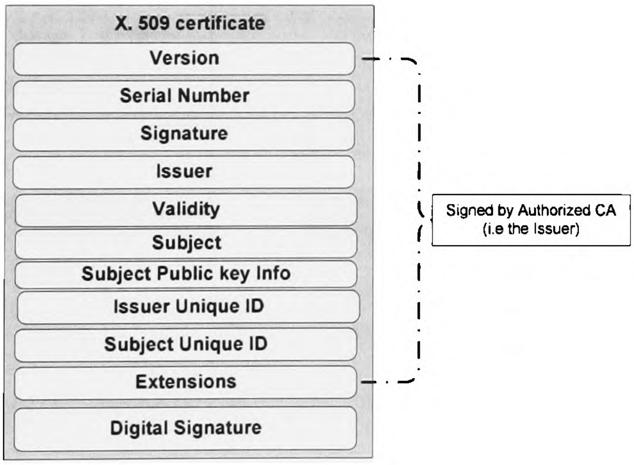


Figure 5.4 X.509 version 3 Certificate Structure

[8] It is not recommended for use by rfc3280

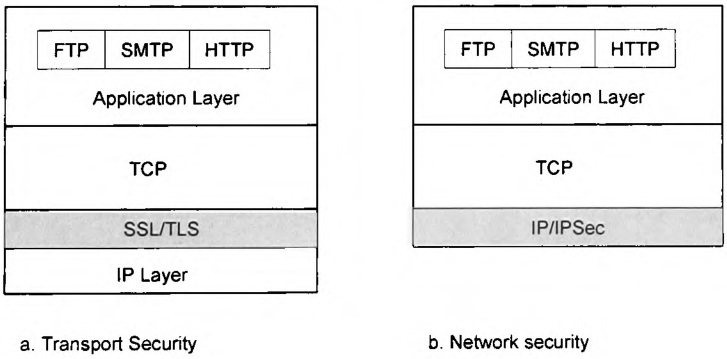
[9] It is not recommended for use by rfc3280

5.5 Network and Transport Layer Security

The interconnection of public networks based on the TCP/IP stack is open and vulnerable to the threat of eavesdropping and traffic monitoring activities. There is, therefore, the requirement for end-to-end secure conversation between communicating devices to ensure confidentiality. The Internet Engineering Task Force (IETF) [145] defined a suite of security protocols that use a range of cryptographic techniques at the network and transport layers of the stack. Figure 5.3 depicts transport and network layers security in a TCP/IP stack, respectively. The sections that follow describe the two main modes of the protocols.

5.5.1 Secure Socket Layer (SSL) and Transport Layer Security (TLS)

TLS defined by RFC2246 and SSL[146] are cryptographic protocols designed to provide a secure pipe between two communication endpoints in the transport layer of the TCP/IP stack. They primarily offer mechanisms to establish secure session parameters, authenticity of communicating devices (server and/or client), as well as data integrity. They use both symmetric and asymmetric cryptography in the management of security between communicating devices during a session. They define a suite of extensible protocols, cryptographic algorithms, and require that communicating participants (i.e. server and the client) are aware of the protocols and algorithms. The SSL (and TLS) makes use of public key certificates to support the services provided, as well as providing the mechanism for establishing trust between the endpoints. Consequently, the reliability of the services offered



a. Transport Security

b. Network security

Figure 5.5 Network and Transport Security Stack

depends on the establishment of trust, or the relying parties may simply wish to trust the certificates at their own risk. The TLS is an advanced variant of SSL with extended features and capabilities.

5.5.2 Internet Protocol Security (IPSec)

The IPSec standard also developed by the IETF provides end-to-end IP layer security that enables private conversation over public networks offering confidentiality, integrity, and authenticity of data communication. It has two main protocols: Authentication Header (AH) and Encapsulation Security Payload (ESP) protocols. These protocols work together using appropriate algorithm(s), and cipher keys to provide high-level end-to-end network security.

The difference between IPSec and SSL/TLS is the layer at which confidentiality is provided between the endpoints. Whilst IPSec provides a secure pipe from the IP layer, so that the source and destination IP addresses are hidden from the public, TLS provides a secure pipe outside the IP layer, exposing both the source and destination IP addresses. Whilst IPSec is suitable for connecting private networks together through public networks, TLS is appropriate for securing communication between two or more connected networks. It is worthwhile mentioning that IPSec and SSL/TLS address confidentiality, integrity and endpoint authentication at the wire level to mitigate against threats posed by man-in-the-middle attacks; and they are not sufficient for addressing application message layer security, particularly in exchanging structured data.

5.6 Conclusion

Information security sprang from the need to make conversations unintelligible to unintended parties. Consequently, privacy enhancement technologies have continued to evolve in an attempt to protect computer and networks, and provide support for preventing undesirable use of the PII. This chapter has demonstrated the availability of rich standard tools, which could provide support for addressing information security related problems. It provides insight into the evolution of important security developments such as cryptography, digital signature and their significance for privacy technologies.

More importantly, cryptography addresses four key information security concerns:

- *Confidentiality* i.e. ensuring that only an authorized recipient can make sense of the information,

- *Authenticity* i.e. ensuring that parties can ascertain the origin of information,

  Integrity i.e. ensuring that information exchanged between parties is not altered in transit and,

- *Non-repudiation* i.e. ensuring that parties are unable to deny their actions associated with an exchange of information.

In sum, this chapter highlights the diversity of these different technologies, and how they address common and sometimes distinct security concerns.

Chapter 6. Relevant Security and Access Control Standards

6.1 Introduction

Security standards provide the platforms for promoting application service interoperability, and they are crucial for ubiquitous computing environments. Utilizing standardized security constructs implies using concepts that have been reviewed by a large community of experts and users. Standards, when widely accepted, can make it easier for users to develop applications that can interoperate with one another. The use of standard security mechanisms is important in order to ensure that the resultant utility [52] of this work can be adapted to real-world applications with minimal effort.

A number of security and access control standards have emerged as guidelines for the design and implementation of infrastructures that could support the prevention of unauthorized access to enterprise resources. These include XML-based security cryptographic services, the eXtensible Access Control Markup Language (XACML), the Security Assertion Markup Language (SAML), etc. Many of these standards are generic, that is, they can support access control requirements in many environments. Consequently, utilizing them in a particular problem domain entails examining them in the context of the identified problems.

This chapter explains the basic concepts and key features of the specific standards, components and architectures that are used in this work. In particular, the chapter gives an overview of eXtensible Markup Language (XML) and its security services, as well as detailed discussion on the XACML and SAML, which play significant roles in the design and development of the infrastructure described by this thesis.

6.2 eXtensible Markup Language (XML)

XML, defined in [147], is an industry standard derived from SGML (ISO 8879), specifically designed to meet the requirements of large-scale electronic publishing. XML is based on a simply structured flexible text format which plays an increasingly important role in the wide range of exchange of data on the Internet. The XML describes a class of data objects called XML documents as well as the mechanism that enables the processing of the XML document. In their normative form XML documents consist of storage units named entities containing either parsed or unparsed data. Typically, parsed data comprises characters in the form of character data, and form Markup which encodes a description of the document's logical structure and storage layout. The XML document is designed to work with a schema or Document Type Definition (DTD), which imposes constraints on the XML document processing and operations. XML has recently received a more widespread acceptance and implementation for the exchange of structured data over electronic networks, particularly among disparate systems. It provides the foundation for XACML and SAML constructs, as well as the new web service framework, which is transforming the dynamic provisioning of distributed Internet services.

6.3 XML Security Schemes

W3C specify two related standards: XML Signatures and XML encryption for creating and processing of digital signatures, and encryption of any digital content (data objects).

6.3.1 XML Signature Scheme

The standard defines a schema for representing the results of digital signature operations applied to any part of an XML document. XML signatures can be used to provide authentication, data integrity and non-repudiation. The core benefits of the XML signature scheme are the ability to sign particular portions of the XML tree; it can also be enveloped, i.e. the signature is over the XML portion of the content containing the signature as an element; and can be detached i.e. the signature is "detached" from the portion of the XML content it signs. Figure 6.1 shows the elements of XML digital signature in a tree format. The root node is the *Signature* elements, which contain the *SignedInfo* element, *SignatureValue* element, *KeyInfo* and Object extension elements. The SignedInfo node contains *CanonicalizationMethod*, *SignatureMethod* and *Reference* elements. The *Reference* element identifies each portion to be signed by a URI attribute, and contains a *Transform* element that specifies an ordered list of processing steps that were applied to the *Reference* portion before applying the digest; the *DigestMethod* refers to the *Hash* function algorithm; while the *DigestValue* contains the value of encrypted digest of the *SignedInfo* element.

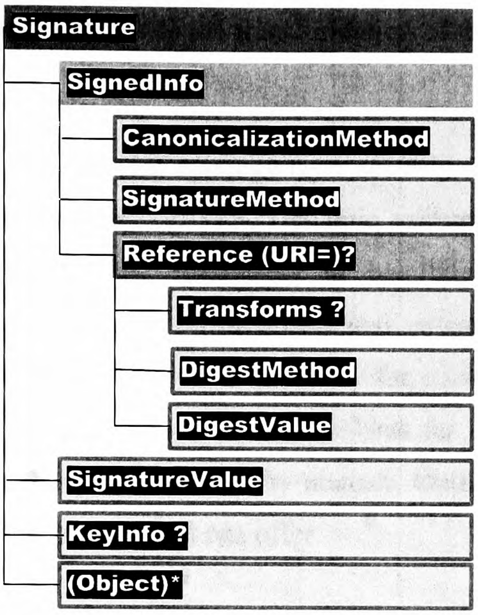


Figure 6.1 Structure of XML Digital Signature scheme

6.3.2 XML Encryption

XML Encryption provides end-to-end message layer security for applications that have a requirement for secure exchange of structured data. The message layer security characterizes an approach where all the information which is related to security is encapsulated in the message. The XML encryption provides the natural way to handle complex requirements for security in most data exchange applications. Figure 6.2 depicts the simplest structure of the XML encryption scheme which can appear anywhere within the XML document structure. The *EncryptedData* element is the root and contains *CipherData* and *CipherValue* elements. The

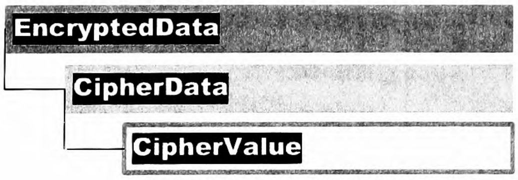


Figure 6.2 The XML Security Service Architecture Scheme

XML encryption syntax provides a standard way for compliance systems to interpret the XML encrypted payload unambiguously. The *CipherData* and *CipherValue* nodes signify that any data within the container have been subject to an XML encryption transformation, and it can contain other information that can facilitate the transformation of the data back to its primitive XML structure.

Generally, XML security schemes provide message layer security that is impossible to achieve with traditional transport layer security i.e. TLS. Dynamic exchange of business information requires some sort of flexibility and robustness in handling structured and complex transactions, whereby secure intermediary processing and extension of messages may be desirable. These security schemes provide the machinery for combining secure and insecure data in the same message payload, which offers mechanisms for supporting the treatment of complex documents dynamically and in trustworthy manner. Outlined below are some of the exceptional message layer security features it can offer.

Partial Encryption of Data: In some application scenarios, it may be undesirable to encrypt all parts of the information; XML security provides a mechanism to encrypt a part of the data and leave other parts unencrypted. It can also handle a complex secure exchange in which several parts of the message can be encrypted uniquely by different entities using different encryption keys. This flexibility also makes it possible to have secure and non-secure data in the same message payload.

Multiparty Encryption: This provides a means to secure sessions between two or more communicating parties simultaneously, i.e. a document can partially be encrypted for multiple users concurrently without compromising security in a more flexible manner.

In summary, the XML and its security services offer significant benefits in addressing security problems in the message layer, and they are the backbone for making the new web services framework secure. In particular, with XML security services, structured data can be selectively signed, or encrypted for those XML elements that essentially require it. These security services are considered relevant to the general development of the framework proposed by this work.

6.4 The Website Privacy Policy and P3P Initiative

It is common nowadays to see a display of privacy statements on websites on how they collect and make use of PII. These statements are usually lengthy legal worded documents [35], often compiled by lawyers, and written in special legal terms [148]. The problem with these policy statements is that most users hardly read and/or understand them. It is not only that they contain legal jargon; an average website user will not spend time reading the document before exploring a site. The Platform for Privacy Preferences (P3P) [148] is an attempt to address the above human limitations by a standardized policy format allowing a website's privacy statements to be uniformly represented and transmitted to user-agents at the instant of service invocation.

The expectation is that compliant user-agents (i.e. web browsers) can automatically retrieve and interpret the service provider's privacy policy statements and match them against the user's published privacy preferences. It is expected that these compliant user agents will help users become more familiar with the service providers' privacy practices, when published in both machine and human-readable formats. Furthermore, it is imagined that it can facilitate the automation of web browser agents in making decisions regarding users' privacy, based on their privacy preferences.

The overall benefit is to lessen the burden of users having to read lengthy privacy policy statements themselves at every site they visit. The P3P privacy statement is based on XML constructs, which describe data collection, purpose of collection, retention period as well as whether information is shared by third parties, etc. The P3P is developed around fair information practice statements [14]. The projection is that in the near future, smart user-agents will be able to predict the behaviour of service providers through an analysis of their privacy statements and adequately inform users. The consequence is that users through their user-agent will have more control over the use of their personal data in an automated fashion. Outlined below are some of the often-stated benefits of the P3P privacy model:

- To promote fair information practices so that commercial sites can imprint trust and confidence in their customers by being transparent in the collection and usage of information.
- P3P based on industry standards can promote openness and improve significantly the level of communication between consumers and websites.
- By being open and making their privacy practices visible, organizations can attempt to take appropriate care of how to collect information that is relevant to them.

- P3P can enable organizations to communicate their privacy enforcement promises to customers.

It is worthwhile mentioning that the P3P platform has created privacy awareness within the community of Internet users and industry stakeholders. In chapter 7, drawn from [149], the applicability of the P3P policy is illustrated in the context of this work. The P3P initiative is the primary Internet privacy framework targeted at addressing a website's handling of privacy in the aspects of collection, storage and dissemination of PII. It has provided some degree of privacy awareness, and become the vehicle for the regulation of online privacy statements through specifications and guidelines. It is apparent that P3P helps service provider websites to convey their privacy statements in machine readable format so that compliant user-agents can compare a site's privacy statement with a user privacy preference.

The P3P framework is more of a guideline than enforcement machinery; it has no apparatus for the monitoring of privacy compliance, i.e. whether service providers actually adhere to their own privacy statements or not. In most cases, the privacy statements have no strong binding to the providing sites, and this implies imposing the user-agent to 'rely' on it or abandon the service access. The lack of strong binding to the providing party makes it vulnerable to abuse and misuse. For example, in a site scripting attack, an attacker can fool the browser by using the right P3P policy, which is genuinely downloaded from the impersonated provider's site. The repercussion is that, since the user has no mechanism to verify and validate the authenticity of the P3P policy statement, the user-agent can easily be fooled. Although it does describe "disputes" and "redress" mechanisms in the event of violations, the fact that there is no verifiable strong binding between the policy and site, can invalidate its use in any privacy dispute. However, although the P3P has not yet been explored to address privacy in access control systems, it is one of the possibilities explored in this work.

6.5 Enterprise Privacy Authorization Language (EPAL)

The Enterprise Privacy Authorization Language (EPAL) [150] is an XML based enterprise privacy language developed by IBM to help organizations' privacy policy writers to define terms, conditions and rules that protect users' PII in compliance with organizational statements on privacy and procedures [151]. Whilst EPAL is not used directly in this work, it is included here for completeness. EPAL has two top-level components that must work together in EPAL compliance systems:

EPAL-Vocabulary that defines the terms that aids the system in interpreting organization-specific privacy elements as well as data that may be required for evaluating other conditions contained in the policy.
EPAL-Policy that defines privacy policy rules that contextualizes what is allowed and what is denied based on a given vocabulary.

The EPAL defines structural elements and hierarchies of data-categories, user-categories, and purposes, sets of privacy actions, obligations and conditions. The user-categories element describes the aspects of entities i.e. users/groups that collect data (e.g. administration department of finance officer). The data-categories element defines different categories of collected data that have diverse privacy characteristics, i.e. medical records, personal details, etc. The purposes element specifies the intended purposes, i.e. administrative purpose (e.g. data is collected for admin purposes), data collected for the processing of annual returns, etc. The actions element indicates how data is used, i.e. disclose versus read. The obligations element defines actions that must be taken in subsequent operations. Figure 6.3 shows the UML representation of the EPAL policy model and the relationships that exist among its various elements. Although the EPAL is a well-defined structural XML–based privacy language, which specializes in data handling at enterprise level privacy concerns, it lacks the flexibility and extendibility to support wide range of use-cases in protecting privacy and confidentiality in distributed environments. In particular, its functionality cannot simultaneously provide support for privacy and confidentiality protection of enterprise-wide access control requirements (i.e. using the same framework for the protection of personal data as well as traditional enterprise resources).

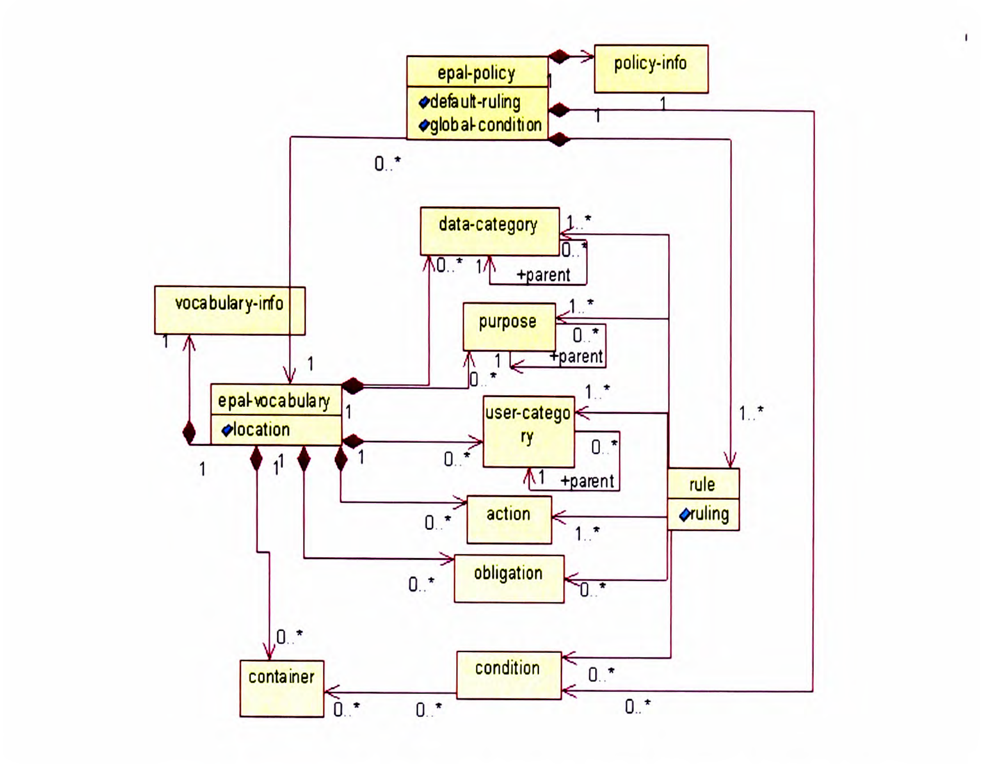


Figure 6.3 EPAL Policy Structure [150]

6.6 eXtensible Access Control Markup Language(XACML) Model

The XACML standard developed by the Organization for the Advancement of Structured Information Standards (OASIS) is an XML-based model that provides generalized functionality for the wider range of access control problems [37]. It is an extensive access control policy language that not only defines the formats and encoding of access control rules, but provides message level request-response mechanisms which allow its distributed components to cooperate in access control operations. Figure 6.4 adapted from [37], depicts the abstract view of the model showing core components and dataflow. It represents a modularized architecture that decouples its subcomponents, based on functionality, in a manner that allows them to be distributed. The following section describes the XACML model actors.

Policy Administration Point (PAP): The entity provides the relevant rules that govern access to protected resources in the form of policy or PolicySet.

Access Requester: The entity that initiates the access control request context that triggers access control operation.

Policy Enforcement Point (PEP): The entity intercepts access requests and enforces appropriate access control decisions. It sends a request to the Policy Decision Point (PDP) through the Context Handler containing the attributes of subjects, resources, action and the environment that describes the ADI of the Target scope.

Context Handler: The component converts the primitive inputs from the PEP to a format consumable by the PDP (request context) and outputs (response context) consumable by the PEP.

Policy Decision Point (PDP): The component makes a decision based on the attributes contained in the request context, against the local policy that governs access to the protected resources.

Policy Information Point (PIP) is the entity that supplies the attribute values required by the PDP including that of the access requester.

Obligation Service: The entity enforces any obligations returned by the PDP after the access control decision.

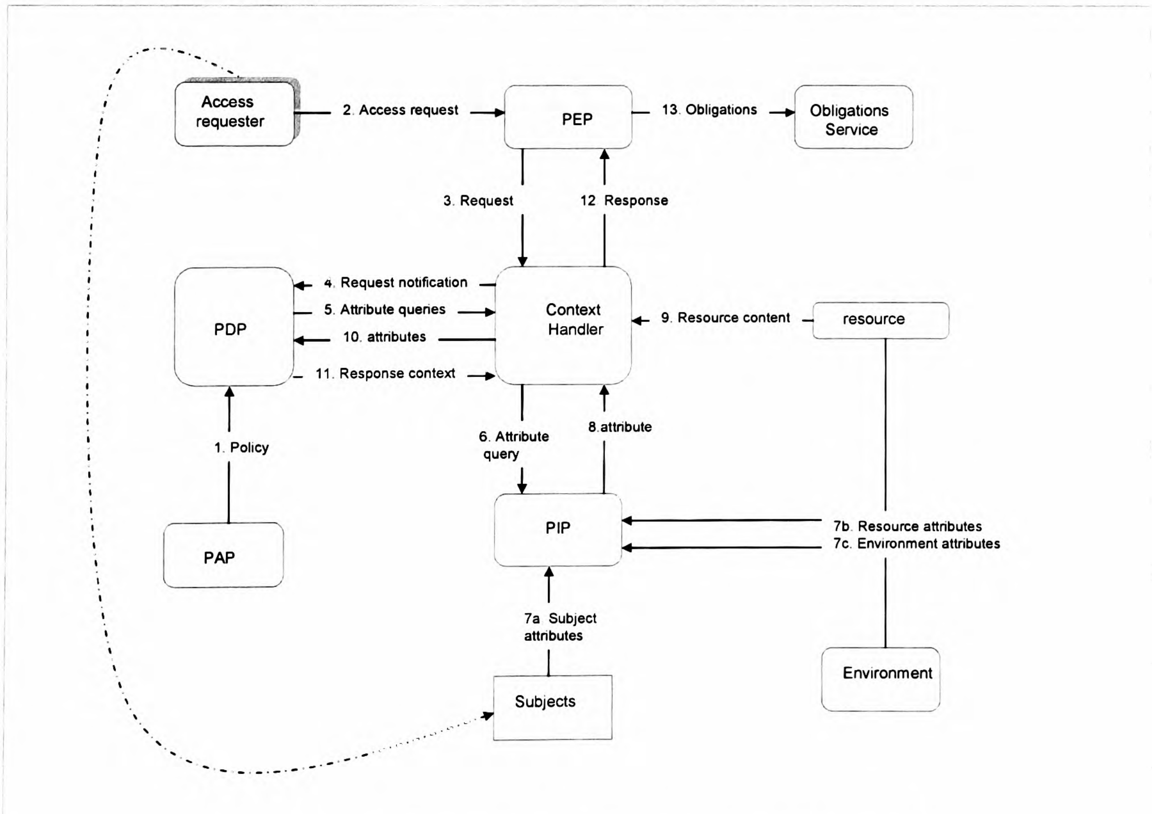


Figure 6.4 XACML Policy Components and Dataflow [37]

6.6.1 The XACML Context

The XACML architecture separates the core access control policy language from the application specific environment, to allow its elements to cooperate even in disparate environments. This contextual abstraction fragments the core XACML and defines an XML schema instance, which specifies the canonical representation of the various access control inputs and outputs. The constructs provide the means that allow XACML attribute-types to be referenced by an instance type, i.e. an attribute can be accessed as an *XPath* expression using the notion of *AttributeDesignator* type defined in the XACML schema, which is an explicit identifier of that attribute. Each XACML primitive attribute type such as *subject, resource, action* and *environment*, together with other optional identifiers, is strongly mapped to a named XACML *Data-type*. The application specific environment takes the responsibility for converting its native attribute representation into a XACML compatible context. This particular functionality makes possible the seamless integration of XACML actors in distributed systems using common semantics and vocabularies. XACML specified two types in its context schema [152] namely:

- *XACML Request Context:* This describes the attributes of XACML request context actors relative to a given policy in which evaluation will take place. In particular, it specifies information about subjects, resources, actions and environment with respect to a particular access decision request consumable by a XACML PDP.
- *XACML Response Context*: It defines elements of the response context, usually from a XACML PDP that is consumable by a XACML PEP. The response context characterises the access evaluation decision result reached by the XACML PDP, and it is represented at top level in the forms of Decision, Status, and Obligations.

A classical example, is a mobile device constrained in terms of computing resources to run a PDP engine, which may rely on a trusted third party PDP to provide an authorization service [149]. The PEP component residing in a Mobile device can send a trusted XACML *request context* to this third party PDP and expects to get back a trusted XACML *response context* to determine access to its protected resources.

6.6.2 XACML Policy Language Model

XACML offers a wider range of access control policy language in several application domains providing the mechanism for a finer access control granularity. At the top-level, it contains three main structural elements considered as a tree: *Rule, Policy*, and *PolicySet*, implying a hierarchical relationship. These elements contain other subcomponents, structured in a style that further describes the detailed properties and scopes of the *Target*. Figure 6.5 shows the various components and typical relationships that exist among them. In the following a brief description of the various elements is provided.

Rule element: This is the innermost element of the three top-level nodes in the XACML language model, and a direct child of a *Policy* element. According to XACML schema [37], multiple rules can be encapsulated in a single policy, provided that they are uniquely identified. The *Rule* element consists of three other child elements described below:

> *Condition element*: This node describes conditional constraints, which further refine the evaluation of an access control context. The XACML standard defined extensive functions and data-types used in evaluating conditional elements and the outcomes are usually TRUE or FALSE. This functionality, in addition to the target scope, provides a fine-grained processing of applicable request context in access control decision procedures. It contains *Apply* element, a native XACML predicate that facilitates the use of extensible mathematical functions to refine the matching and evaluation of a policy context. Figure 6.8 shows structurally, the main policy elements and their hierarchical relationships, which are critical to the processing of a XACML r*equest context.* For simplicity, some sub-elements and attributes are omitted in the diagram, but are discussed later in this section.
>
> *Effect:* The element that describes the intended decision of a policy rule and defines two attribute values: *Permit* or *Deny.*

Target element: Defines the *Target* scope of the policy in terms of *resources, subjects, actions* and *environments* that is applicable to an instance of a *Rule* node. It is important to note that the *Target* in the *Rule* node has a hierarchical relationship with the *Target* in the parent *Policy* node.

Resources element: This element contains a *disjunctive sequence* of *Resource* elements, which describes a set of ***resource***-related entities that can be matched against *attribute* values in the XACML *context request* element *Resource* with the embedded *attribute* values.

Subjects element: This element contains a *disjunctive sequence* of *Subject* elements, which describes a set of *subject*-related entities that can be matched against *attribute* values in the XACML *context request* element *Subject* with the embedded *attribute* values.

Actions element: This element contains a *disjunctive sequence* of *Action* elements, which describes a set of *action*-related properties that can be matched against *attribute* values in the XACML *context request* element *Action* with the embedded *attribute* values.

Environments element: This element contains a *disjunctive sequence* of *Environment* elements, which describes a set of environment-related entities that can be matched against *attribute* values in the XACML *context request* element *Environment* with the embedded *attribute* values.

Policy Element: This node can contain any number of Rule nodes, and its Target node is a filter to the Rule elements. In addition, it contains a unique identifier called *PolicyId*, rule-combiner algorithm identifier, set of obligations and *Target* instance. The *Target* in the policy container defines the semantics of sets of allowable attributes of *resources, subjects, actions* and *environments* within the *Policy* scope. In practice, this feature provides the means of filtering allowable *Target* domains that the *Rule* container can provide. The idea of *Target* at each branch of the XACML tree is for performance enhancement designed to speed up access decision operations.

Rule-combining Identifier: A *Policy* can contain any number of *Rule* elements. To avoid conflicting results, a rule-combining scheme provides the mechanism to deal with and resolve any conflicting evaluation result when multiple Rules must combine to a single deterministic result. In this regard, only one instance of a rule-combining algorithm referenced by its identifier is allowed in a Policy container.

Obligation: The node that describes further constraints that must be enforced by the PEP entity in addition to the deterministic access decision outcome.

PolicySet Element: This is the outermost node in the XACML Policy tree, which can contain any number of *PolicySet* or *Policy* elements. It contains other attributes such as unique identifier, combiner algorithm identifier and annotations. The idea of *PolicySet* is to accommodate enterprise wide policy sets, which can be defined by different security policy administrators, but can be combined into one effective applicable policy.

6.6.3 Combining Algorithms and Policy Evaluation

The combining algorithm is fundamental to the processing and evaluation of a XACML request context. The idea is that individual *Rules,* and/or *Policies* can be combined unambiguously into a single applicable policy in a given access control request context. That is, a combining algorithm defines procedures that allow the evaluation of set of *Rules* and/or *Policies* to obtain a deterministic authorization decision. Figure 6.6 depicts the logical

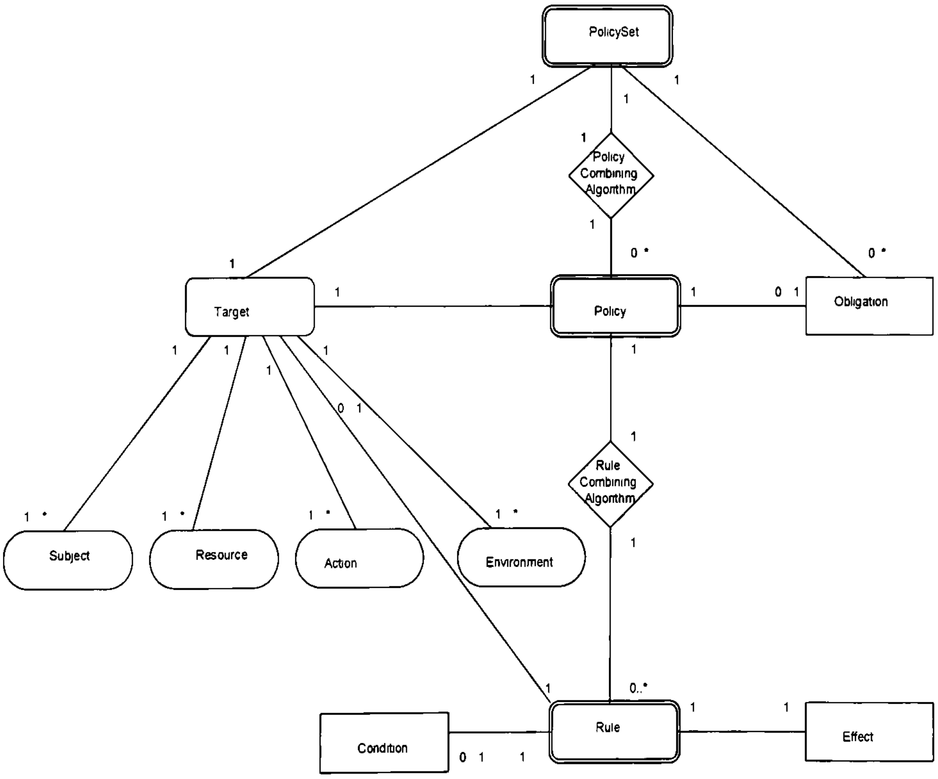


Figure 6.5 XACML Policy Language Model

PolicySet combining model demonstrating how two Policy elements that are contained in a *PolicySet* can be combined into a logical applicable policy. Similarly, figure 6.7 demonstrates the logical combination of two *Rules* elements contained in a *Policy* into effective policy that can be evaluated to obtain a single deterministic decision. Outlined below are some of the native combining algorithms supported out of box by the XACML model:

Deny-Overrides: Where multiple *rules* or *policies* are evaluated, if there is a *Deny Effect* on any of the results, then the final authorization decision defaults to *Deny.* Contrary, where any *rule* or *policy* evaluates to *Permit,* and "all other *rules* evaluates to *NotApplicable*" then the authorization decision shall be *Permit.*

Ordered-Deny-Overrides: In the case of *Deny Effect* attribute, a pre-defined order of precedence defined in the *rules* or *policies* must be followed to arrive at a final authorization decision when multiple rules or policies are evaluated.

Permit-Overrides: Where multiple *rules* or *policies* are the case, if any *rule* or *policy* in the set results in *Permit*, the final authorization decision defaults to *Permit.*

Ordered-Permit-Overrides: Similar to the ordered-deny-overrides, the order in which the

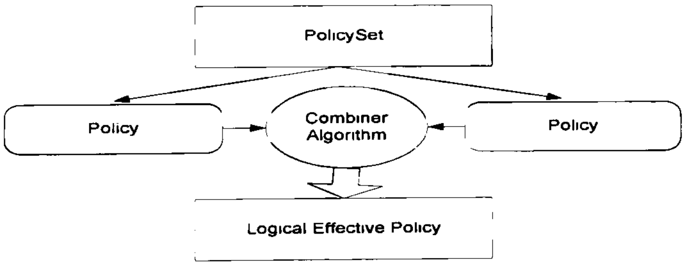


Figure 6.6 PolicySet Combining Logical Architecture

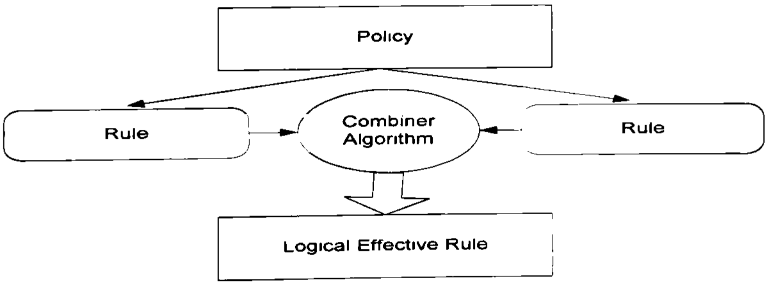


Figure 6.7 Policy Combining Logical Architecture

policies or *rules* apply is pre-defined in order of precedence within the rules or policies.

First-Applicable: This scheme defines the order of applicable precedence; the result of the first relevant *rule* or *policy* determines the final authorization decision regardless of whether it is *Deny* or *Permit* effect.

The combining algorithms mechanisms offer significant benefits and conveniences in the combined treatment of privacy and confidentiality in a large scale deployment.

6.6.4 Policy Indexing

In a typical enterprise access control environment, it is likely that multiple independent policies can be specified to govern access to specific resources. To speed up the retrieval of the applicable policy and verification of its validity, the *Target* element has native support for two main indexing mechanisms. The first approach is to configure the PDP engine to search for applicable policies in a database system at runtime. An alternative approach is to initialize the system to load policy instances at the PDP start up time. The choice of a particular technique depends on performance requirements and application environment.

6.6.5 XACML Data Types

The XACML standard specifies a wide range of *data types* as the basis for creating the predicates for conditions and target matching. It consists of derivatives from pure XML *data types* usually represented as string and the XACML object defined strong typed *data types*. The *data types* are fundamental to the PDP decision making as it converts various string representations into primitive *data-types* for the purpose of matching and evaluation. For example, X.500 directory name, an ITU-Rec.X.520 Distinguished Name described in IETF RFC 2253 "Lightweight Directory Access Protocol version 3 is represented as urn:oasis:names:tc:xacml:1.0:data-type:x500Name in the Attribute *DataType* scope. In the case of the RFC822 name, an electronic mail descriptor is represented as urn: oasis: names: tc: xacml: 1.0: data-type: rfc822Name Attribute *DataType*. Similarly, IP address is represented as urn: oasis: names: tc: xacml: 2.0: data-type: ipAddress which depicts an IPv4 or IPv6 network address and optionally may include the port number. More detailed XACML data types can be found in [37]

6.6.6 XACML Functions

The XACML standard describes standardized native functions for the various matching and evaluation operations within the XACML context environment and is critical to the way PDP performs its primary operations. These functions are identified by unique identifiers which enable the PDP to understand and select applicable data-types. For example, the language specifies a set of equality functions that apply to a range of domains including Boolean-equal, String-equal, Integer-equal, Date-equal specifically designed for the processing of XACML native predicates.

6.6.7 Analysis of XACML Policy Workflow

The XACML model is designed to cover a wide range of access control domains, but provides the techniques that make it easy to solve specific access control requirements. It offers significant flexibility in terms of access rule expressions as well as in policy writing, and this is fundamental in solving complex access control problems [30]. A thorough understanding of the policy structure, syntax, and the underlying schema is critical in analyzing a number of ways it can be applied to address complex security problems. As shown in figure 6.8, the *PolicySet* node is considered as a tree containing one or more children: *PolicySet* or *Policy* and a *Target.*

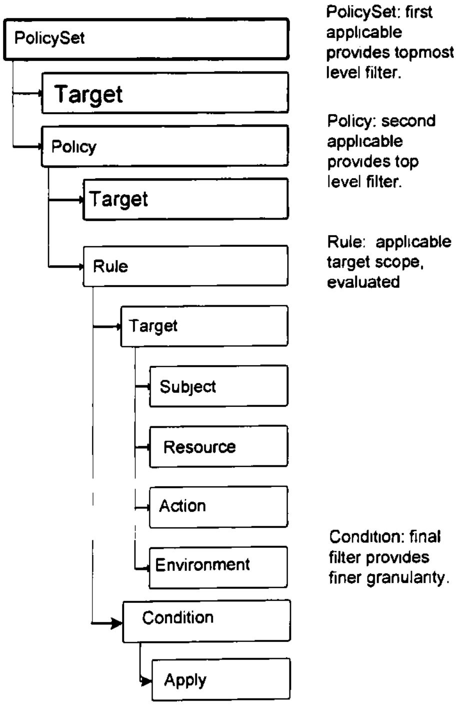

PolicySet
Target
Policy
Target
Rule
Target
Subject
Resource
Action
Environment
Condition
Apply
PolicySet: first applicable provides topmost level filter.
Policy: second applicable provides top level filter.
Rule: applicable target scope, evaluated
Condition: final filter provides finer granularity.

Figure 6.8 XACML Target Scope Workflow

Similarly, a *Policy* contains one or more child elements: *Rule*, and a *Target.* The *Rule* contains the *Target* and *Condition* elements that filter the policy context and provide a finer granularity.

In the XACML schema, each element <Target> contains a description of four core elements which are *subjects, resources, actions* and *environments.* In the *Target* scope, a conjunctive sequence of <Subjects>, <Resources>, <Actions> and <Environments> can possibly be expressed within the extent of <Target> applicability to a request context. The implication is that in processing, there MUST be at least one positive match between each section of the <Target> element and the corresponding section of the <xacml-context: Request> instance. The outcome is that each *Target* at any node of the tree is an intersection of *Targets* in the path that leads to that branch of the tree. In processing an access request, if the *Target* at any top level evaluates to FALSE, evaluating that branch of the tree becomes needless. However, the cascading of *Target* is a performance enhancement tool essentially for filtering a request at any node before advancing to the branches. The effectiveness is actually in the ability of the policy writer to intuitively describe the *Target* scope in each node appropriately to filter out undesirable requests. Further descriptions of inner elements of *Target* child elements are detailed next.

The <Subjects> tag contains a disjunctive sequence of <Subject> elements, which contain a conjunctive sequence of <SubjectMatch> elements. The <SubjectMatch> is used to cover the applicable identifying set of subject attributes that describe the entities permissible within the <xacml-context: subject> element of the request context.

The <Resources> tag contains a disjunctive sequence of <Resource> elements, which contain a conjunctive sequence of <ResourceMatch> elements. The <ResourceMatch> identifies applicable resource attribute types that can be matched against the attribute values contained in a <xacml-context: resource> element of a request context containing relevant attribute values.

The <Actions> tag contains a disjunctive sequence of <Action> elements, which contain a conjunctive sequence of <ActionMatch> elements. The <ActionMatch> is used to cover an applicable set of action-related attribute types that can be matched against attribute values contained within the <xacml-context: action> element of a request context.

The <Environments> tag contains a disjunctive sequence of <Environment> elements, which contain a conjunctive sequence of <EnvironmentMatch> elements. The <EnvironmentMatch>

describes environmental rules which can be used to filter out subject entities within the <xacml-context: environment> element of the request context.

Additionally, each of the above designated *Match* elements may contain the following sub elements:

The *MatchId* attribute element that describes the applicable matching function and its type and is often represented as anyURI type.

The <xacml: AttributeValue> tag contains the embedded attribute value of a particular Target child entity such as the Subject.

The <AttributeDesignatorType> tag contains the necessary information that identifies one or more applicable attribute value types in the *Target* scope of a *request context.* This primitively permits the use of named attributes in which respective *AttributeId, DataType* and *Issuer* properties can be matched against those of the *request context* using defined URI equality functions. The <AttributeDesignatorType> contains AttributeId (required), DataType (required), Issuer (optional) and MustBePresent (optional) attribute elements.

The *<AttributeSelectorType>* tag identifies one or more attribute value types in the *Target* scope based on an x-path expression.

The *<Condition>* tag plus its child element <Apply> provide the mechanism for finer processing of the access *request context*; the idea is to filter undefined subject attributes in the *Target* section, by expressing additional environmental constraints that further refine the access evaluation, which must take place before the final access decision takes effect.

6.6.8 Request Context Evaluation

Generally, in processing an access request, the PDP evaluates the local policy based on the properties of the *Rule* element against the incoming request context attribute properties. Critical to the processing operations are three components: *Target, Effect* and *Condition* components within a local policy domain. The processing of the <Target> at the parent (outer) node and the <Target> at the inner or child node is done in two ways. In one approach, the <Target> element of the outer node <PolicySet> or <Policy> is processed as the union of all the <Target> elements of the referenced <PolicySet>, <Policy> or <Rule> elements of the inner node. In the second approach, the <Target> element of the outer node is processed as the intersection of all the <Target> elements of the inner nodes. The outcomes of the two approaches are different; in the first approach, the <Target> element of the outer node is only applicable to any decision request that matches the <Target> element of at least one of the inner node components.

In the second approach, the <Target> element of the outer node is only applicable to the decision requests that match the <Target> elements of every child component. This unique functionality can intuitively be employed to address complex access security decisions involving different policy administration points within an enterprise. For example, a privacy framework should make space for the unhindered free flow of information for legitimate purposes. Whilst the domain administrator can express a generic rule governing access to information flow, the individuals can in addition specify their privacy access policy; at runtime these two policies can be combined in a *PolicySet,* and using an appropriate combiner algorithm, access operations can reach a deterministic decision without conflicts.

6.7 Security Assertion Markup Language (SAML) Standard

SAML is an Organization for the Advancement of Structured Information Standards (OASIS) specification developed around XML constructs to provide message level security, essentially for communicating identity and access management information across autonomous boundaries [107]. It provides standardized generic syntax and processing semantics to represent the client authentication, claims, entitlements, and attribute properties in a well-structured xml format. That is, assertions made about a network entity by another entity, called the SAML Authority, or attesting party to a consuming (or requesting) entity, called the relying party. It describes both the assertion structure, set of request – response protocol messages, processing rules, and various profiles for practical implementation to support other XML based technical security standards.

The model plays an important role in message layer security and is central to the rising Federated Identity and Access Management, an emerging platform for the promotion of interoperability among business parties. It offers robust security architecture for seamless inter-domain business interactions offering an insulating buffer for disparate systems to share identity attribute-information. Behind the SAML components is the schema which defines the structural elements and types that make up the assertion block.

6.7.1 SAML Components

Figure 6.9 depicts the architecture of the SAML model showing the various components described in the following section.

SAML Assertion: An assertion makes a statement about a named principal or attributes of an entity, confirming that the establishment of security context about an entity has been done by a particular means, at a set time, and this is uniquely attested by a SAML Authority or the issuer. The SAML Assertion can contain any of the three components explained below:

Authentication Statement that is created by an asserting party about a SAML event, which makes the statement of fact that an entity has authenticated successfully, and includes the means of authentication, time, and other processing information.

Attribute Statement that contains specific identifying attributes about the principal or the holder of the assertion.

Authorization decision statement that describes what the authenticated entity is allowed to do, plus the privileges, entitlements or capabilities, which can attest to the claim.

SAML Protocols: The SAML request-response protocols define the XML construct for the exchange of assertions, and include the following:

Authentication Request Protocol that describes a means to allow a relying party to request assertions containing authentication information and optional attribute statements.

Single Logout Protocol that specifies the means that allows simultaneous logout of active SAML sessions of a particular entity across relying domains (i.e. service providers that made use of the assertions).

Assertion Query and *Response Protocol* that defines a set of query and response protocol messages, particularly how a relying party can ask for assertions of an authenticated subject.

Artifact Resolution Protocol that provides a way to pass protocol messages by a reference using a small. fix-length value, called an artifact. The relying party uses the *Artifact Resolution Protocol* to ask the providing party to dereference[10] the artifact and return the

[10] In programming parlance, accessing a value referred to by a reference is called dereferencing it.

message value. Two channels are used here: the resolution request takes place over SAML HTTP binding, and the response over synchronous SAML SOAP binding.

Name Identifier Management Protocol that provides mechanisms to manage the principals name identifier or value, and terminates a name identifier between the identity provider and service provider.

Name Identifier Mapping Protocol that provides a means to programmatically map one SAML name identifier into another provided it is compatible with predefined rules.

SAML Bindings explain in detail how SAML protocol messages can be conveyed using existing application layer protocols.

HTTP Redirect Binding that describes the use of the HTTP Redirect mechanism to communicate SAML messages.

HTTP POST Binding that explains how SAML messages can be used within the encoded base64 content of an HTML form control.

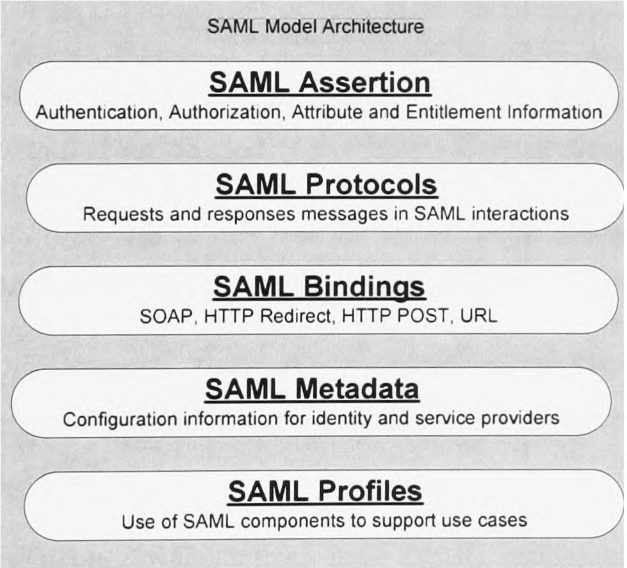

SAML Model Architecture
SAML Assertion
Authentication, Authorization, Attribute and Entitlement Information
SAML Protocols
Requests and responses messages in SAML interactions
SAML Bindings
SOAP, HTTP Redirect, HTTP POST, URL
SAML Metadata
Configuration information for identity and service providers
SAML Profiles
Use of SAML components to support use cases

Figure 6.9 SAML Model Architecture

HTTP Artifact Binding which defines how the message sender and receiver can use either HTML form or URL query string to convey an artifact.

SAML SOAP Binding that describes how to use SAML messages within SOAP over HTTP.

Reverse SOAP (PAOS) Binding which defines the Enhanced Client and Proxy mechanisms that allows an HTTP client to act as a SOAP responder.

SAML URL Binding that defines how to resolve and retrieve an existing SAML assertion based on a uniform resource identifier (URI).

SAML Profiles: This mechanism defines how SAML components achieve interoperability by combining with other existing standards such as XACML, X.509, etc.

Web Browser SSO profile which describes how the Authentication Request Protocol, SAML Response Messages and Assertions combine to provide a single sign-on with standard web browsers using HTTP Redirect, HTTP POST and HTTP Artifact bindings described above.

Enhanced Client and Proxy (ECP) profile that defines how specialized clients or gateway proxies can use a particular profile with PAOS and SOAP bindings explained above.

Identity Provider Discovery Profile that describes one convenient way for allowing a relying party to find out the identity providers a user has already visited.

Single *Logout Profile* that defines how the SAML Single Logout Protocol is executed by the use of SOAP, HTTP Redirect, HTTP POST, and HTTP Artifact bindings.

Assertion Query/Request Profile that describes how SAML entities can leverage the SAML Query and Request Protocol to retrieve SAML Assertions over a synchronous binding, e.g. SOAP.

Artifact Resolution Profile defines how SAML entities can leverage the Artifact Resolution Protocol over a synchronous binding, i.e. SOAP to get the protocol\messages referred to by an artifact.

Name Identifier Management Profile that explains how the Name Identifier Management Protocol could be used with SOAP, HTTP Redirect, HTTP POST, and

HTTP Artifact bindings.

Name Identifier Mapping Profile defines how the Name Identifier Mapping Protocol leverages a synchronous binding, i.e. SOAP binding.

6.7.2 The SAML Assertion Structure

SAML Assertion structure [107] is a purely a XML-based encoding scheme that represents one or more statements made by a SAML issuer or SAML authority. The high-level outer structure is depicted in figure 6.10 showing that of an Assertion Statement. In figure 6.10, the *Issuer* element indicates the name of the asserting party or SAML authority; the *Signature* element indicates the signature block which contains information including the public key, algorithms and transforms, digital signature, etc; the *Subject* element indicates the identity of the holder; the *Condition* element indicates constraining conditions, and the *Assertion Statement* can indicate any of the assertion instances including authentication, attribute, authorization decision, or other user-defined specifics.

6.7.3 SAML Notion of Privacy

SAML v2.0 defines a generic privacy mechanism anticipated to provide communicating party privacy protection. It primarily uses an opaque pseudo-random identifier to shield an assertion holder from being explicitly identified by a relying party. Additionally, the pseudo-random identifier is designed to inhibit collusion between multiple SAML providers in an assertion session. Furthermore, SAML specified other security mechanisms, i.e. metadata for sharing federation meta-information; encryptions formats, and attribute profiles, etc. The SAML components, messages protocols and profiles provide the essential services which business

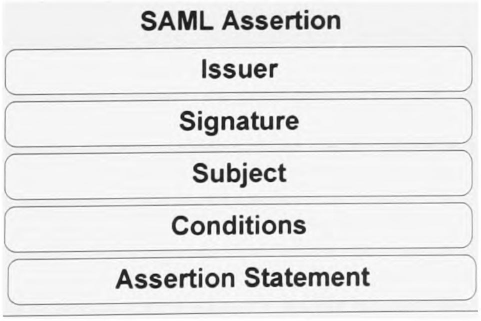


Figure 6.10 The SAML Assertion Structure

partners in distributed environments can leverage to secure their messages. More information regarding SAML can be found in [153].

6.8 Web Services Security Standards (WSSS)

The WS-Security standards [154] also defined by the OASIS, aim to address security enhancement in SOAP message contexts, and a variety of extensible message level protocols are defined. They describe approaches that autonomous business partners with different security requirements can leverage to secure web services interactions in terms of message content integrity and confidentiality. These allow web services actors to associate and broker trust establishment, identities and claims for the purposes of authentication and authorization. These standards mainly focus on the following:

- Support suitable sharing of identity, claims, tokens, privileges, etc., and multiple security tokens formats.
- Communication of authentication and authorization information using similar or distinct mechanisms, but abstracted at a high level to provide a common platform for interoperability.
- Brokering of trust and security token exchanges: use of intermediaries to broker authentication and authorization wherever desirable whilst maintaining high quality security, and support for multiple trust domains.
- Optional obscuring of identity information and other attributes to enable privacy.

WS-Security provides mechanical approaches to ensure that business partners using web services in unsecured public networks can interact without the ability of a third party to view the messages, and are able to determine the originality of the messages as well as ensuring that the messages were intact whilst in transit. In addition, OASIS defines WS-Trust models that describe various trust establishment approaches and infrastructure as further necessary components in securing web services interactions.

6.9 The Liberty Alliance

The Liberty Alliance is a consortium of major IT stakeholders that develop guidelines and specifications expected to assist in the development of interoperable Federated Identity aware systems [99] to solve non-functional business process problems. The goal is to develop effective and efficient frameworks that will govern how identity related information is shared among communicating business partners. Federated identity is about linking users' digital attribute accounts issued by identity providers with service providers who consume the identity information. This particular account linkage underlies the service advocated by this institution. It facilitates users' experience in the use of SSO to move around the federated sites or services without the requirements to sign on each site again. It specified a suite of standards, specifications, recommendations and guidelines that core Liberty compliant systems should comply with.

The Circle of Trust (CoT) specification deals with contractual aspects of "rules, policies, obligations, procedures, risk and remedies" that will oversee the Federated participants including individual users [155]. Its privacy and security best practices describe the guidelines and recommendations in terms of privacy and security which it expects Liberty enabled applications to comply with [93]. In spite of these extensive guidelines, it recognized that compliance is the responsibility of application providers, as it has no mechanism to monitor, audit or enforce compliance.

6.10 Conclusion

The XML is an emerging important business tool revolutionizing the exchange of information across disparate systems. It is the foundation for many new internet security standards, which include the XACML, SAML, P3P, EPAL, etc. The XACML and SAML are two important models underlying the development of the infrastructure proposed in this thesis for supporting privacy and confidentiality protection. This chapter described the essentials of these security standards, their encoding schemes and capabilities. More importantly, the analysis of the XACML policy instances is given in the context of this work, describing the various alternative approaches in the evaluation of XACML policy rules. Overall, these standards form significant input to chapters 7 and 8.

Chapter 7. Building Privacy and Trust into an Access Control Framework

7.1 Introduction

The web technology has somewhat changed the way Information Systems (IS) interact today; as a consequence, traditional approaches to system development [57] have altered as well. Driven by these new business environments, organizations have a need to expose some part of their enterprise services to the outside world. This implies greater demand to allow access to enterprise computing operations whilst ensuring security and privacy. Addressing privacy and security issues in this context necessitates a requirements gathering for the application development phase. Given the multiple level of interactions between participating parties in distributed electronic transactions, requirements analysis is a critical phase, which covers the complex tasks of deconstructing how a system should behave, as well as helping to expose the system properties, or attributes. Thus, the need for a thorough system requirements gathering to guide the software development life cycle phases is imperative.

This chapter examines the generalized view of distributed access control, and briefly reviews authentication, authorization and trust in the context of the distributed XACML model. It describes a security threat model based on a typical e-procurement use-case in an attempt to define the scope of the applicable privacy and security requirements. This systematic approach is expected to aid in identifying the principal actors, and their relationships in distributed access control interactions. Furthermore, the chapter will also appraise trust models and various options for establishing trust relationships in distributed environments. Additionally, the chapter deals with the policy framework, detailed description of the WS-XACML profile, architecture and usage. Lastly, a conceptual design supporting this work resulting from the inputs made in previous chapters, and the analyses carried out in this chapter is presented.

7.2 Generalized Access Control Context

Conceptually, the stages in distributed access control are performed in two steps: authentication and authorization. Some assumptions are necessary to model this scenario. It will be assumed that in typical distributed environments, authentication should be handled as a separate service at the clients' security domain and authorization at the service provider's domain. A typical access control flow is shown in figure 7.1, demonstrating how an application specific access filter, say PEP, can interact in support of authentication and authorization processes. The access filter in this sense can be seen as a service request filter, which determines which resources need access control, and whether authentication and/or authorization are required before access is granted or denied. In order to avoid unnecessary authentication and/or authorization, the access filter can use subcomponents such as AuthnProxy (Authentication Session management) and AuthzProxy (Authorization Session Management) respectively as shown in the diagram to ensure that authenticated and/or authorized requests are not asked to perform these operations again in the same session.

Figure 7.1 shows that authentication and authorization services are operated in different autonomous security domains, entailing requirements for trust establishment that may inherently be complicated. In order to ensure privacy protection, the initiating client may be unwilling to supply to the PEP all the required subject attributes before service invocation. Additionally, the authentication phase validates the client's origin, providing the PEP some attribute information that is passed to the PDP; meaning that to complete the access decision operation, the PDP must have access to the remainder of subject's attributes to complete the decision process. But since the PDP makes the request through its Context Handler to an external entity, trust must be established. This additional request for attribute information from an external entity triggers privacy because the external entity must ensure that the requestor will treat the attribute information with some regard to privacy. Figure 7.1 clearly shows the complete data flow from the client initiating a service request to filtering of the request by both the authentication and authorization services when required. The next section attempts to put the developed understanding into a distributed XACML interactions context.

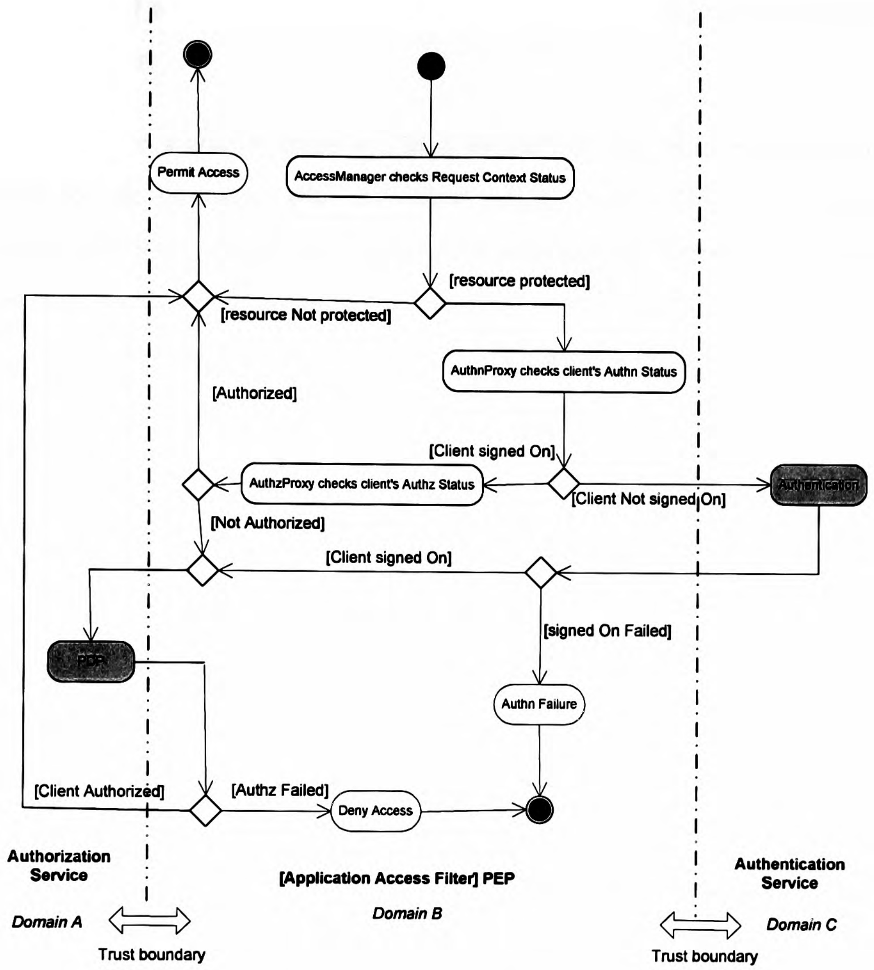


Figure 7.1 Conceptual Access Control Flow

7.3 Trust Context

In the section that follows, trust establishment is examined to further deconstruct and clarify the trust boundaries. Trust is fundamental, in distributed transactions, trust can be established between people and people, people and services, and services and services, and may demand handling the trust relationships dynamically [46]. Figure 7.2 depicts a simple dialogue between two parties: Alice and Bob in a typical trust context and is described as follows:

- Alice approached Bob's door and knocked;
- Bob asked 'who are you';
- Alice responded 'I am Alice;

- Bob said come in, because he recognised that was the voice of Alice and expected to see her.

Analyzing the above scenario, there is a high probability that Bob may open the door and see instead of Alice, an impostor, who mimicked Alice's voice. This naïve example can be the basis to justify the two variables associated with trust namely: behaviour and expectations. The illustration implies an innate issue of risk elements in the general concept of trust; buttressing the need to build systems that will allow communicating parties to negotiate trust based more on other properties than the PKI relationships can provide.

In the digital world, trust models are the basis of verifying and validating trust relationships, claims, privileges, properties, identity-information, etc. giving the relying party the choice of whether to trust a providing party or not based on certain defined rule constraints. Traditionally, the certainty to trust an entity can be based on the characteristics of the trust model in place.

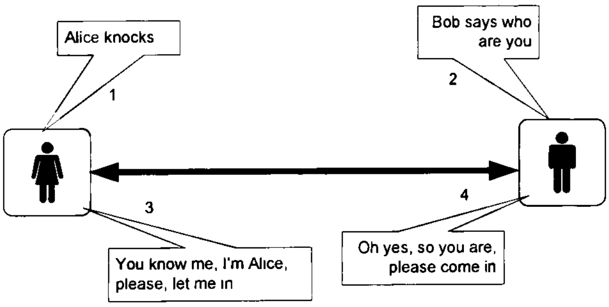

Alice knocks
1
Bob says who are you
2
3
4
You know me, I'm Alice, please, let me in
Oh yes, so you are, please come in

Figure 7.2 Typical Trust Dialogue

7.3.1 Direct vs. Indirect Trust

There are two basic trust relationships: direct and indirect (sometimes referred to as transitive trust). In a typical access control operation, the service gatekeeper needs to verify and validate the claims made by a client; these operations require some sort of trust relationship. Direct trust is based on shared knowledge or a shared secret, such as username/password pair, PKI certificate, etc. which is usually established out-of-band between parties prior to communication interaction. In the case of indirect trust, a party needs to validate the claims made by another party but there is no direct trust relationship between them. In other words, the parties require a trust broker, or trusted third party that both of them can trust, in order to communicate and share sensitive information. Figure 7.3 depicts two basic primary trust models showing direct and indirect trust relationships.

Figure 7.3 (i) depicts the idea of direct trust between a service client (SC) and a service provider (SP) that requires a proof-of-possession to establish trust. In contrast, in indirect trust models, a third party has to vouch for the identity or attribute claims, privileges, properties etc. of a party. This is illustrated in figure 7.3 (ii).

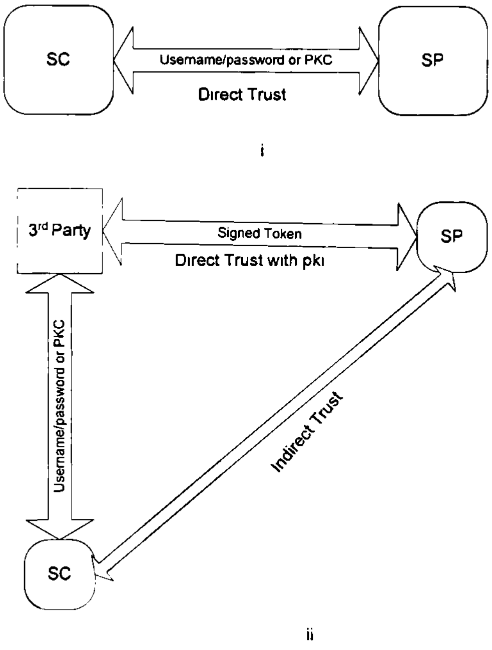


Figure 7.3 Basic Trust Model

In figure 7.4 (i), a more sophisticated trust model is depicted showing boundaries of security domains, as well as some kinds of possible trust relationship. Figure 7.4(ii) demonstrates the steps and interactions that a SC can use to request a service from an SP, when they do not have direct trust relationships. In this mode, the assumption is that direct trust between the providing party and relying party is impractical, so intermediaries, trusted by both parties must intervene.

1. The SC authenticates and obtains a signed token from its local Identity Provider (IdP) /Security Token Service (STS).

2. The SC initializes and constructs another authentication request using the token from step 1 to an IdP/STS trusted by the SP. The SC's IdP/STS and SP's IdP/STS need to have prior trust relationships for this protocol to succeed. The SP's IdP/STS validates and processes the token request, issues a fresh token or cross certifies the token issued by the SC's IdP/STS, and returns it to the SC.

3. The SC constructs a web services message with the token and requests a service from the SP. The SP validates and processes the request and sends the appropriate response to the SC.

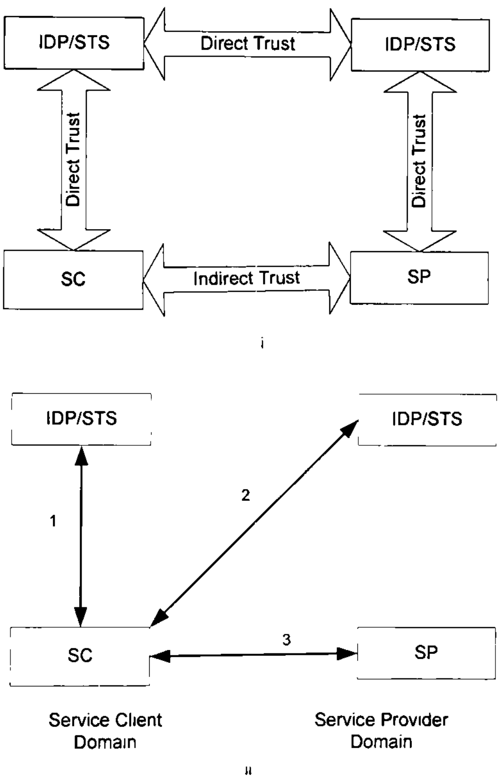


Figure 7.4 Brokering of Trust via Two TTPs

It can be seen that the basis of trust between the SC and the SP can be analyzed as follows. The direct trust between the SC and its IdP/STS exists in the form of a username/password pair. In contrast, the SC has no direct trust with either the SP or its IdP/STS. Thus, trust between the SP and SC, which has no direct path, is provided by the trust between the two IdP/STS respectively. It can be seen that from the SP's viewpoint, the SC's IdP/STS has no direct trust relationship with it, but it can trust the assertion claims based on its relationship with its

IdP/STS. This trust model exposes some inherent security flaws, which can be exploited by a greedy participant in high risk business transactions. The recipient party merely trusts the assertion claims made by a third party on the identity of the holding party, which are unlikely to reveal all the intentions of that party (i.e. behaviour). Arguably, this trust establishment cannot solely be relied upon to create confidence in an entity, but can provide the foundation upon which parties can negotiate and build more trust, through the mutual exchange of what they expect from the other party, and what they are willing and able to do for that party.

On the basis of the above understanding, it can be concluded that in distributed environments where multiple actors exist in multiple domains, it is essential that brokered trust establishment be utilized as an 'introductory trust', upon which a higher threshold of trust can be built by the parties themselves. Having said this, it is equally vital to point out that what will determine the trust ascribed to a party might be based on some measurable risk metrics, which are outside the scope of the present work. To sum up, it can be concluded that privacy assurance cannot sufficiently be based on the provision of PKl relationships only, but requires more dynamic exchange of obligating and binding constraints between the parties themselves as the basis for building a higher level of privacy trust. Nevertheless, PKI still has an important role to play.

7.4 XACML in a Distributed Context

The XACML overviewed in chapter 6 confined its scope to access control model and language constructs, and intentionally overlooked how XACML actors in distributed environments can collaborate in a mutual access control interactions. The assumption is that it can utilize other complementary standard models to describe assertions, protocols or transport mechanisms that will enable its distributed actors to converse in secure trusted manner. In practice, the PEP entity is responsible for protecting access to the resources. The PEP filters every access request and applies appropriate enforcement, which may include establishing the authenticity of the request or sending the description of the request to the PDP entity in the form of XACML context request. Two things can be established here. First, in the case of authenticity, it may require a formal authentication of the initiator by any available means. Second, the PDP needs to evaluate the request against its available policies and attributes to make an authorization decision, which has to be passed back to the PEP

Traditionally, XACML PEP obtains a description of the request context from a range of possibilities including distributed (perhaps on-line) *Attribute Authorities* (AA) or *Attribute Repositories* (AR) for the subject designator attributes. Similarly, the PDP may interact in distributed manner to obtain the policies from PAP or from *Policy Repositories*. Furthermore, the PEP may not necessarily supply all the subject's attributes at the instance of making a request context from the PDP. In this case, the PDP through its Context Handler component will attempt to augment the subject attributes by asking a designated AA or AR for other attributes. This particular convenience is what makes XACML a candidate for protecting privacy and confidentiality in distributed environments. Of course, this raises the issue of trust; the distributed components must trust each other. Where the components exist in autonomous security domains, the level of trust required may vary, and even become complex. It is therefore important to examine how XACML actors can interact in distributed environments which will help in understanding the rest of the sections.

7.4.1 Distributed XACML Context Interaction

Figure 7.5 shows XACML actors in a distributed environment and their interactions which are described below: (It is assumed that the client has successfully authenticated with its local authentication service and token(s) issued in that respect in the form of assertion)

1. The client constructs a service request with the token containing the information about the subject and sends it to the PEP; it is important to note that because of privacy concerns, not all the subject information is included in this phase.
2. The PEP constructs a request context containing the token as the subject descriptor, and obtains other information, i.e. properties of the resources, time constraint, etc, and presents it to the Context Handler;
3. The Context Handler formats the request context appropriately and presents it to the PDP to decide whether access should be allowed;
4. The PDP obtains all applicable policies and evaluates them against the request context. If the PDP cannot complete its operation because of missing subject attributes, it requests the missing attributes from the Context Handler;
5. In turn, the Context Handler requests the missing attributes from the client's PIP (It is assumed that the client performed authentication with its PIP before service invocation). The dotted line indicates this external conversation;

6. The client's PIP returns the missing client's attributes to the Context Handler;
7. The Context Handler sends the returned attribute information on to the PDP;
8. The PDP completes the evaluation process and passes decision results to the Context Handler;
9. The Context Handler formats the decision results into a response context and sends it to the PEP;
10. The PEP interprets the response context and enforces a decision by either allowing the requested resources or indicating that access is denied to the client.

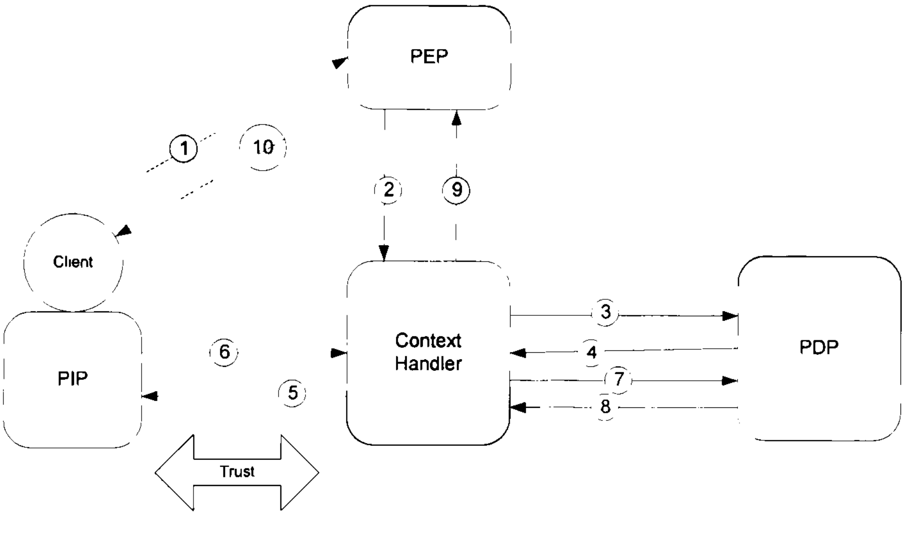


Figure 7.5 XACML Components Interaction Context

Analyzing the above interactions, it becomes apparent that the steps in the dotted lines require some form of trust to be established, to ensure privacy and confidentiality. In the scenario where the client has to reveal more attribute-information to a remote party before the access control decision is taken, privacy becomes a serious issue. The interactions above involve communication between external entities in different security domains, increasing the need for privacy, confidentiality and trust in distributed environments. This implies that for adequate privacy protection, negotiations between the external entities are desirable. This is necessary to allow both parties to determine how, where and when to reveal resources and attribute information, and apply desirable obligating constraints on the other party to guarantee privacy and confidentiality. In this regard, it is important to mention that trust is the vehicle for achieving these goals. Based on the above assessment and knowledge, the following necessary assumptions are made[11]:

- No access is allowed to a protected resource by default. The access rules for a protected resource must require attribute information of the client making the request before access is allowed, otherwise access is denied. Where more than one attribute is required by the access control operation, the rules shall be expressed in a manner such that one of the subject descriptors provides the initial mechanism to establish the first degree of trust context.

- The client may not be willing to disclosure applicable attributes of privacy concern at the first stage of the access request. In this regard, the authentication phase between the client and its local authentication provider, in which a token is issued, partially reveals information (i.e. origin authenticity) about the client, to serve the above-mentioned first degree of trust context.

- The initial information provided by the client is not sufficient to breach the client's privacy or the confidentiality of the protected resources. This potentially defeats any attempt by a bogus participant to mount probing attacks, often associated with trust negotiation [8, 123]. The underlying theoretical assumption is that if the interacting parties decide to withdraw from the transaction at this stage, they are not overtly exposed to privacy and confidentiality risks. Moreover, if any of the participants is an imposter, then the initial access rule filter will screen out the request, and the imposter will not succeed in any subsequent interactions. Although it can be argued that having a clue about the origin of a party is a privacy risk, what is important is the privacy risk impact factor and its actual consequences, which are outside the scope of this thesis.

- Where the first degree of trust establishment described above is insufficient to gain access to protected resources, both parties require other levels of trust establishment to reach their various goals. In this scenario, any attempt by the service to request

[11] Authentication is not the primary focus of this work as most of the existing authentication approaches can be integrated with the described framework.

more of the client's attributes will trigger mutual trust negotiations before sensitive information is exchanged.

- To make this negotiation phase privacy compatible, the service simply sends its access control policy as requirements across to the client [149]. The client, uncertain whether the service will respect its security preferences, cannot reveal sensitive information, but can respond with a similar counter policy. This iterative process triggers privacy trust negotiation and exchange of relevant attribute information, which can take a number of rounds until both parties are satisfied to release their various sensitive resources.
- It is assumed that the above scenarios do not guarantee assurance that the parties will respect each other's privacy, so additional steps are needed; this prompted a strong consideration of a workable protocol that will enable communicating parties to generate and exchange difficult-to-repudiate tenable evidence about their contextual information, in order to provide end-to-end privacy and confidentiality.

To further the above suppositions and substantiate them in the proper context, a security threat modelling technique is utilized to critically survey and scope the privacy problems in an application environment [87]. The benefit of security modelling is to ascertain the extent of security to apply in a given application domain, through a proper analysis of inherent and foreseeable security vulnerabilities and threats, and determine suitable mitigations. In the next section, this approach is used to investigate the casual effects of privacy and confidentiality in an access control environmental context in an attempt to validate the earlier assumptions made about privacy and confidentiality.

7.5 Security Threat Modelling and Analysis

Threat modelling is a formal approach that attempts to uncover application level security threats and vulnerabilities to determine the possibility of risk thresholds [59]. In the investigation, three factual elements, (i.e. privacy, confidentiality and trust) are the variables central to the study. It is important to deconstruct them subjectively to distinguish their characteristics, and further put their relationships in the proper context. For this purpose, the following definitions are assumed.

- *Confidentiality* is that notion concerned with making sure that only an entity with the right privileges gains access to protected resources.
- *Privacy* is that notion of ensuring that the legitimate entity that has gained access to protected PII treats the PII trusted to it with respect to the providing party's security preferences.
- *Trust* is that means to establish the confidence that a resource consuming entity will act in a predictable and/or expected way.

Analyzing the above definitions, prevailing causal assumptions underscore the probability that a legitimate entity may have access to controlled resources, but abuse them by using the resources for other purposes than those originally stated. This phenomenon could be intentional or unintentional; whichever is the case, the potential exists for a privacy violation, bringing anticipated threats into focus. In retrospect, it can be deduced that confidentiality ensures that parties with the appropriate level of access privileges can gain access to restricted resources; but after the access, what they do with the resources has to be addressed by privacy mechanisms. In the privacy context, no subsequent use of attribute information other than for the originally stated purposes is a contractual obligation that must be respected by parties.

Trust on the other hand is the element that focuses on expected behaviour, i.e. the expectation that the communicating parties will act mutually and compatibly without incurring risks to each other based on the trust threshold provided by their properties or attribute-information. Furthermore, this brings the requirement that in distributed transactions involving two or more autonomous security domains, more security constraints are necessary for effective resource control, since requirements can rarely be static. This suggests that authorization and trust establishment have to be treated dynamically, as remote enforcement of obligating constraints is more exigent. To validate the above empirical assumptions, a use-case based on a classic e-procurement service within the construction industry is modelled. The objective is to capture the variables from various interactive steps and deduce successive message flows in order to determine the likely threats to privacy and confidentiality.

7.5.1 The e-Procurement Use-case

During the procurement phase of a construction project, the main contractor initiates a process aimed at ordering the products, materials and components essential for the construction of the building project. The contractor defines his product needs and publishes a call for tender at a dedicated web service portal aimed at potential product suppliers. The establishment of the call triggers a bidding process, and product suppliers can access the portal to search for calls appropriate to them and make offers, based on the publisher's requirements and other security constraints. Subsequently, the contractor can retrieve all the offers, analyze and rank them accordingly to determine suitable offer(s) before placing a purchase order.

Shown in figure 7.6 is the basic architecture illustrating the three main participants, possible trust relationship boundaries and typical flows of messages. In the above procurement scenario, certain transactional and security characteristics have to be identified to facilitate the modelling and analysis of the security threats. From the architectural point of view, the following assumptions can be made:

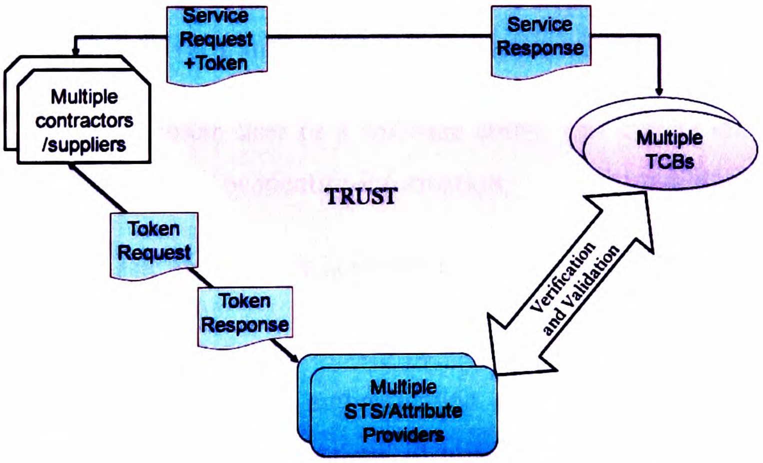


Figure 7.6 Roles in e-Procurement Use-Case Architecture

- Three types of actors exist, namely; the contractor or supplier, the Security Token Service/Attribute Authority (STS/AA) and the portal services-Tender Call Broker (TCB), with each playing a distinctive but sometimes similar role in separate interactions.

- The contractors and suppliers can act as service clients, and in some instances implicitly as service providers, and have similar characteristics in terms of service interactions.

- The TCB and STS are trusted third parties, and can belong to a particular construction consortium and/or geographical area, but can be multiple and/or federated. The TCB is an intermediary or service discovery broker, which provides a service interface on behalf of the suppliers and contractors, and is governed by a set of defined rules and procedures. This is to facilitate administration of tenders and biddings. Additionally, the TCB is a platform that provides the federation for trust establishment among participants.

- Contractors and suppliers may not necessarily have a previous trust context before the invocation of services or belong to the same TCB and/or STS.

- The participants can exist in multiples with or without the existence of direct trust relationships, but mutual trust should be established before they can exchange sensitive resources.

- The various participants may have properties and/or identity-information that requires privacy and confidentiality protection.

- Either a human user or a software entity, can initiate the process, and has similar properties and/ or identity-information.

In figure 7.7 the basic architecture is shown plus the underlying steps involved.

1 The client prepares a service request message with a suitable software application. It initializes and presents an authentication request to its local authentication provider - STS/AA. The client presents an identifier or proof-of-possession in the form of a username/password pair to perform this phase.

2 The STS/AA authenticates the client's claim(s), to verify and validate its identity or confirmation that the client has successfully authenticated with another trusted broker (if in a federation). The STS/AA can determine whether to issue a security token based on the local policy, and if the client belongs to a particular role i.e. membership role, the STS/AA issues a security token and passes it to the client.

3 The client packages the web service message with the token and makes a service request to the TCB portal. In the case of a contractor, it is attempting to publish a call-to-tender, whereas the supplier would be attempting to retrieve some tender calls.

4 The TCB portal through its security handler sends a validation request or asks for more attributes of the client, to determine the access rights of the requesting client.

5 The STS/AA processes the validation request or attributes request and presents an appropriate response to the TCB.

6 The TCB validates the STS/AA response, completes the client's request and sends an appropriate response to the client. For example, in the case of a supplier attempting to retrieve calls, it needs to match the request against the advertised policy of the contractor that placed the call, and determine whether this supplier can be allowed. A contractor may place certain constraints on potential suppliers, which can act as initial filter, i.e. the supplier must possess membership of certain consortium and a proof of annual turnover of a certain amount. On the other hand, a supplier may place similar obligating constraints on the contractors, i.e. validity period of bids (in privacy terms: maximum retention period). Some or a subset of these contractual obligating constraints can be advertised in the web service policy, if desirable.

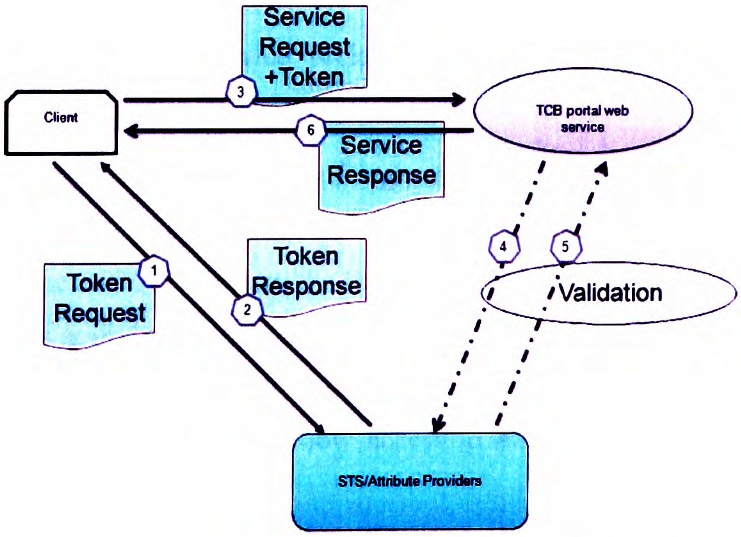


Figure 7.7 Use-case actors' Interactions

In practice, a contractor client retrieves the bids, analyzes and selects the one that best suits its criteria and places an order for the goods. It is expected that the above steps would be followed by either the contractor or supplier, and the TCB must ensure that the contractor retrieves only the call-to-tender it has advertised.

The above interactions give rise to some empirical deductions and understanding, which can be summarized as follows:

1. The interactions involve client initiators making a service request to protected services that require access control measures. The resource release is governed by access control rules that determine who does what and when. The owners of the resources may advertise their complete policies or subset of their policies and other obligating constraints.
2. Several assets of the actors are involved and may require privacy and confidentiality. These include conventional resources; participants' attribute information; meta-information; and contractual business level information.
3. The actors may not all share a common security domain, so trust establishment is a critical factor in the overall interactions, and is paramount to the security of the web services conversations.
4. An actor can place obligating constraints on a participating party, and should be able to say what it is able and willing to do for the other party.
5. The TCB is a service broker governed by an enforceable set of rules and procedures.

Figure 7.8 shows a simple Data-Flow Diagram (DFD) in the context of XACML distributed actors, and gives a detailed description of the flow of messages from one XACML actor to another. Here, the STS/AA replaces the PIP. The above understanding exposes the fact that the client initiator is scared to submit all attribute information pertaining to the request at one go, so it considers leaving out sensitive attribute information in the initial service invocation. In contrast, the service is unable to allow access to the initiator without the complete set of

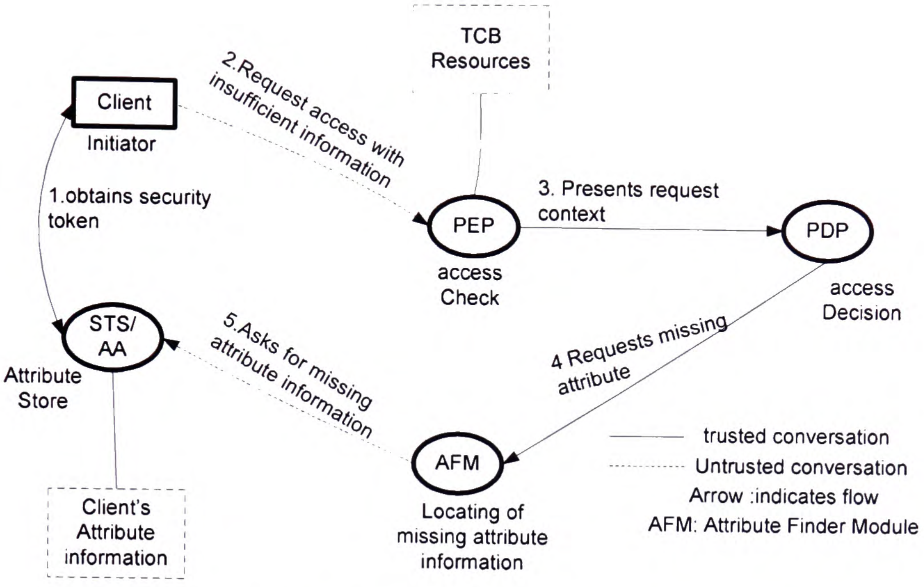

TCB
Resources
Client
Initiator
2.Request access with insufficient information
1.obtains security token
PEP
access Check
3. Presents request context
PDP
access Decision
STS/ AA
Attribute Store
5.Asks for missing attribute information
4 Requests missing attribute
AFM
Locating of missing attribute information
Client's Attribute information
trusted conversation
Untrusted conversation
Arrow :indicates flow
AFM: Attribute Finder Module

Figure 7.8 Data-Flow Diagram (XACML Distributed Context)

attributes that will satisfy the access rules. Decomposing the DFD, security vulnerabilities[12] and threats can be identified particularly within the untrusted interactions. Outlined below is a summary of identified threats in the context of the use-case with respect to privacy and confidentiality.

Undesirable Information Disclosure: It may be desirable to restrict some calls-to-tender to certain groups of suppliers or consortia. The bids need to be kept secret until the close of the call; in this case, the suppliers' offers are vulnerable to undesirable exposure i.e. a supplier entity may require that its bid be handled with utmost confidentiality, and not disclosed to competitors before the close of the call. For example, a malicious contractor can retrieve a call-to-tender it did not publish, or a supplier may gain access to a competitor's offer. From the viewpoint of privacy principles, requirements that can be deduced from these scenarios include the notions of *use limitation. choice/consent, etc.*

Tampering: The tendering process may be vulnerable to unwarranted manipulation by malicious participants. A malicious contractor or supplier gains access to the published call-to-tender or bids and modifies them. In privacy terms, this is simply a data security issue.

[12] Here, security vulnerabilities are considered purely in the context of the thesis; other inherent security vulnerabilities are assumed to have been dealt with in other related work.

Repudiation: A malicious contractor or supplier performs an action that cannot be traced back to them i.e. a supplier makes a lower offer in order to win a bid, and later denies making the offer. Furthermore, a party obtains PII and discloses it to a third party which cannot be accounted for in the case of privacy breaches. From a privacy perspective, accountability principles relate to the repudiation security property, which implies or supports the earlier assumption of the need for remote enforcement of privacy obligations.

Elevation of Privileges: A malicious contractor or supplier performs actions it has no privilege to do, i.e. a contractor retrieves bids for a call it did not publish, or a supplier makes a bid that it is not qualified by default to bid for.

Privacy Support: A participant's attribute identity information or meta-information or business information, i.e. memberships of a consortium, price of goods, may be vulnerable to undesirable privacy threats. For example, a malicious party who has access to a supplier's profile can place unjustifiable restrictions on the supplier, which potentially excludes the supplier from making bids (discrimination threat mentioned in chapter 3). Information obtained legitimately by parties can be vulnerable to unwanted disclosure, misuse or abuse. The underlying privacy consequences have been dealt with in chapter 3.

Trust Context: The various participants require some sort of trust relationship to be established. The reliability of the different interactions depends on the form of the trust established and the ability of a recipient party to accept the claims made by a providing party. Since the process is brokered by a third party, it is potentially vulnerable to trust relationship breaches. The participants may have to rely on the assertions of a third party STS/AA on one hand, and a TCB on the other, as the basis of the trust. The degree of trust establishment that may be satisfactory in certain high-value transactions depends on the form of trust mechanisms available. The assurance that it will all happen within mutually and acceptable practices, is a major trust concern.

The analysis of the above in terms of security requirements exposes the fact that privacy protection is tightly associated with confidentiality and trust, and as such, requires that they be treated simultaneously. The causal findings complement the assumptions previously made concerning privacy and confidentiality. Often, confidentiality is used as a substitute for privacy, but it has been established that they are not identical, and it is important to accurately differentiate them in order to identify the associated challenges and risks. Arguably, unlike confidentiality, privacy has contractual properties and obligations that are backed up by legal framework as well as FIPs. Moreover, this emphasises the need for privacy guarantees and enforcement of the guarantees when transactions span across autonomous security domains. This is supported by the earlier argument that a party that may have legitimate reasons for the possession of PII, and may as well store and use it subsequently without owner's knowledge and consent. This means that where there are no strong binding obligating constraints between communicating parties, privacy may be overly violated. Based on the above critical appraisal, outlined below are security requirements that can be deduced:

- The TCB should have a fine-grained access control to clear or screen requestors for security or reliability. The TCB must enforce appropriate policy rules and a statement of practices to be followed to ensure compliance with relevant security requirements. Doing this may require the combination of a TCB's site policy with the service owner's policies, i.e. a contractor's policy, to ensure that appropriate security preferences are enforced at runtime, whilst ensuring that the policy does not contradict or hinder the legitimate free flow of information.

- Participants may want confidentiality of their information. For example, a contractor entity may specify a pre-qualification a potential supplier must meet, in order to screen out some categories of acceptable suppliers by defining certain cut off criteria.

- Participants' privacy: the various participants' information requires privacy preservations. The supplier entity may place restrictions on what the contractor can do with its bids, e.g. validity of bids has the privacy characteristics of 'maximum retention period', such as the number of days a bid is valid for. Participants may have various service level agreements that require strong privacy bindings, i.e. disclosure to third parties, choice/consent before information can be used other than for the originally stated purpose.

Given the above security requirements, the exchange of some of this service meta-information between parties needs to be handled dynamically as business requirements are expected to change regularly. This strengthens the earlier argument that the trust provided by PKI may not be sufficient to guarantee the remote enforcement of privacy.

7.6 Overview of Proposed Infrastructure

In traditional access control, the main actors can be described as the *Initiator*, the *Gatekeeper*, and the *Target*. Figure 7.9 depicts the symmetrical infrastructure showing the systems distributed components. The XACML Trust Management Infrastructure (XTMAI) in the diagram acts as the protected resource's gatekeeper; being symmetrical, each XTMAI can function at both ends whether in IdP or SP, to enable mutual interaction between them. The XTMAI component provides unified security in a common model allowing the XTMAI aware services to treat both privacy and confidentiality in a well understood manner. As earlier mentioned in this chapter, the authentication phase between the initiator and its local authentication service provides the initial trust context upon which higher level of trust can be built.

However, the means by which the initiator performs authentication at the local provider is outside the boundary of the present work, as existing authentication methods can be used. Nevertheless, from the trust establishment perspective, the authentication service must have an existing trust relationship or operate in the same runtime environment as the client's attribute provider. The initiator must use this as an entry point where privacy negotiation is an important

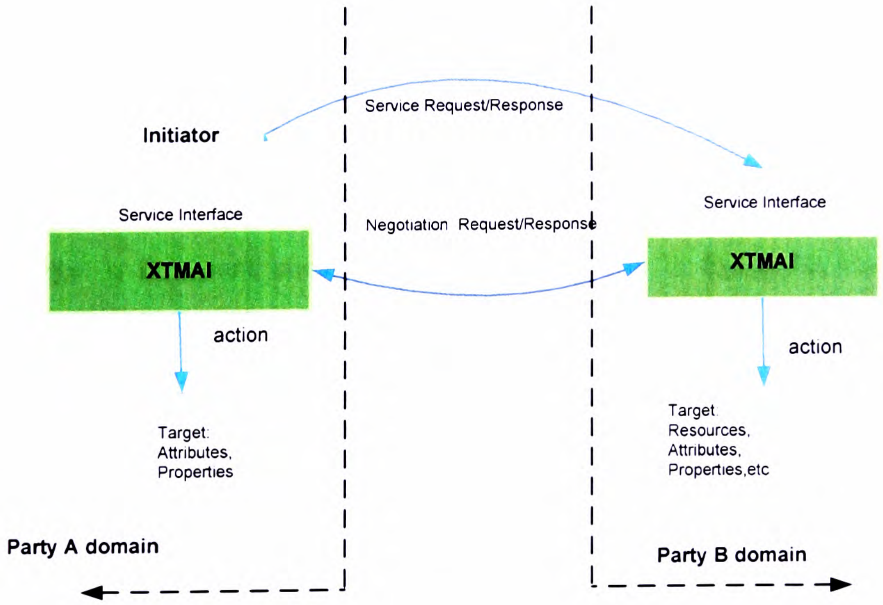


Figure 7.9 The XTMAI Symmetric Infrastructure

security requirement. Following this, an overview of the participants in the architecture is presented. The details of this XTMAI architecture are given in chapter 8.

7.6.1 Initiator

The initiator is an entity that kicks off the interaction, such as a human user or a software component; whichever is the case, the properties remain the same. The primary factor is that the initiator must obtain a trusted provable identity, or possess certain privileges or properties that form part of the access control decision information. The Initiator in this infrastructure can be seen from two entry modes: service request-response mode and trust negotiation request-response mode. The link between the two modes is the initial trust context used by the initiator in the service request-response entry mode. In practice, the service initiator has to provide an authentic trust context; as mentioned earlier, this may be obtained by authenticating with an appropriate authentication service trusted by the other entity.

7.6.2 The XTMAI

This is the gatekeeper entity that guards the *Target* based on the policy constraints made available to it. It handles all trust access control operations on behalf of the *Target,* using the available policy rules and other environment conditions. It has two modes, the service mode for service request-response, and a trust negotiation mode for privacy trust negotiation interactions. A trust context established in service request-response mode couples and synchronizes the two interactions.

7.6.3 Target

The *Target* is the assets and/or resources under guard. In figure 7.9, the *Target* in party A domain refers to the clients' attribute information, whilst in the party B domain, it refers to traditional resources plus other privacy sensitive business information.

Given the overview of the proposed infrastructure, it is crucial to take a look at the policy framework that will ultimately drive the privacy trust access control systems described by this thesis. The next section discusses such a policy framework.

7.7 Access Control Policy Framework

The essence of access control management systems is to restrict access to a *Target* based on pre-defined rules or policies. In effect, the policies guard the systems against unauthorized access and modifications, whilst ensuring their availability to users with provable claims. Enforcement of access control requires that all access to a system and its resources be governed by well defined rules, which describe the characteristics of the native actors, such as *subjects, objects*, and *actions* involved in the access control operations. Already there are a number of policy and meta languages, which can be utilized to specify access control rules [37, 151, 156, 157].

In practice, there are two main parts fundamental to describing access control rules: *separation of duties* and *least privilege principles*. Whilst *separation of duties* deals with the notion of the distribution of tasks and associated privileges for particular business operations among multiple roles or users, *least privilege principles* address the aspect of granting no entity greater access to a *Target* than its job function demands. It is important that the access control language model provides a normative way to support the basic principles of privacy and confidentiality. Based on the threat model covered in this chapter, listed below are factors that rationalized the choice of a policy framework.

- *Expressiveness*: Support for privacy and confidentiality in one suite is a primary requirement considered by the thesis. Overviewed in chapter 4, existing access control systems overlook the simultaneous treatment of privacy and confidentiality. Thus, a language that naturally provides finer-grained access control rules to capture both privacy and confidentiality requirements is desirable.

- *Extensibility*: Business requirements can change as well as associated security requirements. A language that has several extensibility points, which can address a range of different use-cases, is advantageous.

- *Semantics*: To support interoperability across multiple independent platforms, a language that is rich in semantics is required for mutual understanding of a package of vocabularies. In access control operations that span autonomous security domains, it is important that what party A thinks party B will give to satisfy its requirements is indeed what party B released, and that party A will rightly interpret what party B released and vice-versa.

- *Monotonicity*: Monotonicity is an important characteristic of a trust negotiation policy language [158]. It is important that the language provides a normative way to enable incremental exchange of policies and credentials that satisfy them, in a manner that granting of additional privileges, if possible, should progress until negotiation succeeds or fails.

7.7.1 The Rationale for Choosing XACML

The factors considered in choosing a policy language are discussed above, and XACML happens to satisfy the essential requirements. The XACML policy model is generic, yet can be tailored to address many complex security rules through its broad capabilities and the ability to extend them. Outlined below is a summary of XACML's appropriateness:

- *Distributed policy administration*: The concept of PolicySet in which a set of policies can be combined at runtime into an effective applicable policy by using a suite of available combining algorithms is excellent for privacy and confidentiality. This empowers departmental policy authors with an organization to express policies for different applications, and at the same time have one super policy that controls all these other policies. As mentioned in chapter 1, there is a need to balance users' privacy preferences against the legitimate free flow of information. For instance, an enterprise or government can set an overriding policy to enforce the unhindered legitimate flow of information.

- *Extensibility*: As mentioned in chapter 4, in open and dynamic environments, security requirements can change significantly. XACML provides scalable points of extensibility, and has been extended in many instances [38, 149, 159].

- *Interoperability*: XACML is a powerful and flexible language, and has since been deployed in a wide range of application environments [128, 159, 160].

- *Expressiveness*: XACML is generic, yet can be used to express many kinds of security policies i.e. an administrative policy, privacy policy, a role-based policy, simple XML predicates, etc. It is a policy language that describes subjects, resources, actions, and other environmental conditions using identifiers that can be mapped directly to primitive access control actors, allowing automated processing and enforcement of authorization decisions. It provides far-reaching support for defining variables, which can permit grouping of attributes, multiple roles, etc.

Taken together, the motivation for choosing this industry standardized model is an attempt to make this work adaptable in real-world applications with minimal effort. Moreover, its significant support for simultaneous treatment of privacy and confidentiality across autonomous security domains is essential and a critical success factor. The XACML model was exhaustively described and analyzed in chapter 6, but to make the proposed policy framework complete in context, the WS-XACML profile [149] will be discussed in the next section.

7.8 The WS-XACML Policy Profile

In a web service environment, a range of specifications have been defined for many aspects of Quality of Service (QoS) including WS-Reliability, Metadata, Transaction, Resource, Security, for the efficient deployment of web services. One such standard is Web Services Security [161], which is yet to address authorization and privacy issues to cover a wider scope of application environments. The WS-XACML profile draft [149] describes a formal way to utilize the XACML model between communication endpoints in web services environments. It specifies formats for four information message types:

- an authorization token or credential for carrying an authorization decision across realms,
- a policy assertion type that is based on XACML elements which can embed WS-Policy or other XML constructs,
- ways to wrap P3P policy preferences and match them using standard XACML evaluation engines, and
- XACML Attributes conveyed in SOAP Message Headers in such a way that they are provable and valid having been issued by a trusted authority.

In some cases, SPs will want potential requestors to know exactly what access requirements are needed in order to invoke their services. In other cases, a web service is unlikely to advertise the full access requirements, where such requirements could reveal important business information if given to arbitrary strangers. In a situation where privacy and confidentiality are security concerns, parties are unlikely to disclose their sensitive resources, without the assurance that their resources will be accorded security protection. But WS-XACML does not indicate how the assertion policies can be used in privacy negotiations or the means to strongly bind the assertions to the providing party. In chapter 8, the problem is addressed, using the Obligation of Trust protocol, which utilizes the capabilities of WS-XACML.

The WS-XACML profile only provides encoding schemes to describe the privacy and confidentiality concerns of enterprise resources and personal information in a symmetrical mutual manner. For example, a service that supports P3P policy can specify the privacy requirements associated with a given resource. A potential requestor may also have various privacy preferences. The requestor may need to ascertain whether the service is able to satisfy its security preferences with respect to a particular interaction, and if so, how the service intends to fulfill any obligating constraints. The profile describes a formal way for carrying XACML policies and/or other XML based security profiles between communicating parties.

7.9 WS-XACML Context

In many situations, a web service will not want to publish the full access requirements, but a subset, which can act as a first-level filter to minimize undesirable disclosure of sensitive information to arbitrary strangers. This fits into scenarios where an SP considers releasing the full access requirements in one-shot to unknown clients to be a business risk, but could incrementally make the access requirements known as more and more trust-levels are gained between it and the client. In fact in some situations, knowing the type of access control policies, and obligating constraints prior to service invocation, can allow parties to carefully calculate the risks associated with such interactions. A mutual agreement on acceptable policy variables or alternatives, and conveying that to a party in a tamper proof manner, can be an assurance that the other party's obligations are understood and can be respected.

In most of these situations, some web services may not have access to the authorization decision policies within the domain, or may simply be running within constrained environments (i.e. a mobile service provider may have less computing resources) where they are unable to run a Policy Decision Point (PDP) engine. In such cases, the devices would rely on some trusted third party PDP to provide a signed authorization token to a requesting party, which indicates that the TTP has evaluated the requester's request against some access policies and certified that the request can be allowed. Additionally, in other cases, it could be that a web service or client would require a potential interacting party to activate a particular role, i.e. '*Liberty Consortium Membership*', '*certified ACCA accountant*', etc. in order to interact. The client or service with the right privilege can request activation from the particular *Role Activation Authority* (RAA) prior to service interactions, and the relying party can then query the RAA to determine whether the role holder actually activated such a role. Where Privacy and confidentiality are considered highly important, parties may rely on the facilities of WS-XACL's *Requirements* and *Capabilities* to advertise their security preferences.

The WS-XACML *Assertion Type* is an abstract framework that describes an entity's Web Service's access control policy in the context of different policy domains, such as authorization or privacy domains. The name of the Assertion's element indicates the domain to which it applies, such as *XACMLPrivacyAssertions* for the privacy domain and *XACMLAuthzAssertion* for the authorization domain. The *XACMLPrivacyAssertion* deals with privacy specific Assertions, which can carry *Requirements* i.e. what the asserter requires of the other party, and *Capabilities* i.e. what the asserter is willing and able to do for the other party if its *Requirements* are satisfied. Figure 7.10 shows the WS-XACML model, which defines a *XACMLAssertionAbstractType.* This allows constraints on a policy vocabulary to be expressed as XACML *Apply* functions. The *XACMLAssertionAbstractType* contains two sets of constraints, as shown in figure 7.11. One instance of this type is the *XACMLPrivacyAssertion,* whose *Capabilities* element describes the *Obligations* that are being accepted and the information that will be provided. The *Requirements* element specifies the *Obligations* that the sender requires of the other party in order to proceed. In figure 7.10 the complete architecture of the profile and its subtypes are shown. Next, the descriptions of WS-XACML components are given below:

WS-XACML: Requirements

The *Requirements* node describes the information or behaviour that the policy owner requires from the other party in terms of a policy vocabulary. The *XACMLAsserttionType* that contains no *Requirements* element indicates that the entity advertising the assertion has no requirements on the other party that it is willing to publish. This node contains other elements described as follows:

Vocabulary: The vocabulary node contains the identifier of a policy vocabulary: a set of policy variables or document schema containing policy related information. There may be more than one vocabulary instance in a *Requirements* node.

XACML Policy: This element describes a native XACML *policy* instance, which can specify requirements with respect to a policy target.

XACML Predicates: The element is a native XACML variable reference, usually expressed in terms of the *Apply* function. There may be any number of such *Apply* elements in a *Requirements* section.

WS-XACML: *Capabilities*

The *Capabilities* node describes the information or behaviour that the policy owner is willing and able to provide to the other party in terms of a policy vocabulary. In effect, it represents information the publishing entity is willing to release, or obligations the entity is able and willing to fulfill in typical web services interactions.

Vocabulary: The vocabulary node contains the identifier of a policy vocabulary, a set of policy variables or document schema containing policy related information.

7.10 XACMLAssertionType Scope

The WS-XACML assertion instance can be viewed from three dimensions as follows:

1 The *XACMLAssertionAbstractType* is defined with respect to policy vocabularies that are specified in the *Requirements* and *Capabilities* sections of the *XACMLAssertionAbstractType*. In processing the assertions, *XACMLAssertionType* SHALL apply only to those specified vocabularies.

2 The *XACMLAssertionType* is defined with respect to one or more specific policy targets. A *XACMLAssertionType*, when included in a WS-Policy instance, represents *Requirements* and *Capabilities* that apply to the targets of the WSP-Policy instance in

which it appears. Other contexts in which a *XACMLAssertionType* is used shall specify how the policy targets associated with the *XACMLAssertionType* are determined.

3 The *Capabilities* expressed in a *XACMLAssertionType* are defined in conjunction with the *Requirements* in the same *XACMLAssertionType.*

The underlying scope that defines the vocabulary is significant in the processing of *XACMLAssertionType*, and two assertions must be of the same type before they can be compared and evaluated.

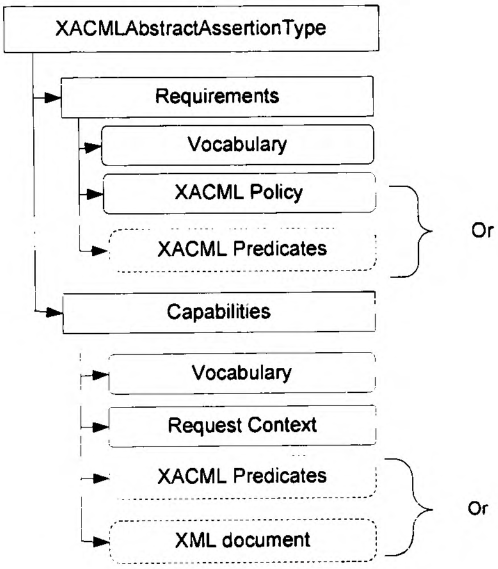


Figure 7.11 XACMLAssertion Policy Structure

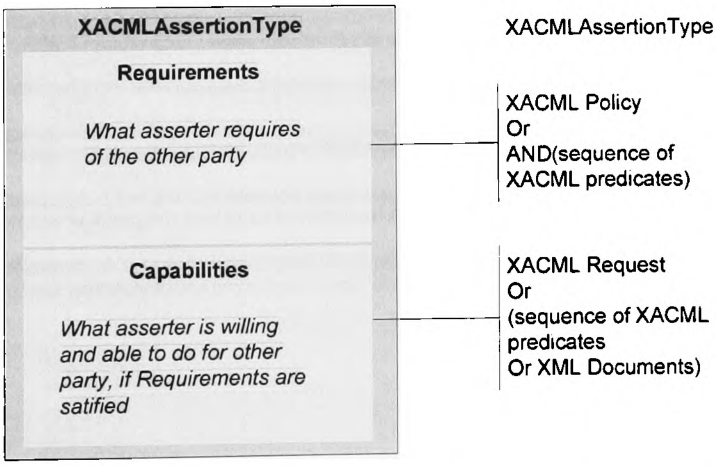

XACMLAssertionType
Requirements
What asserter requires of the other party
Capabilities
What asserter is willing and able to do for other party, if Requirements are satified
XACMLAssertionType
XACML Policy
Or
AND(sequence of XACML predicates)
XACML Request
Or
(sequence of XACML predicates
Or XML Documents)

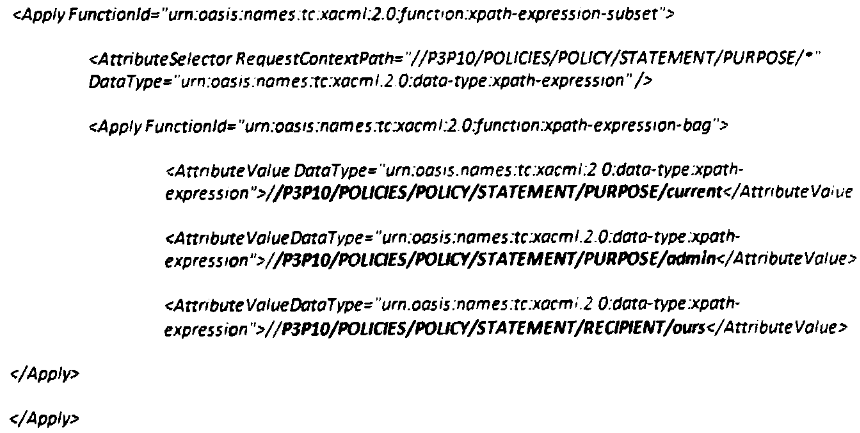
Figure 7.10 WS-XACML XACMLAssertionAbstractType

7.11 The P3P XACML Mapping

Section 5.8 described a P3P policy framework without mention of how it can be integrated with XACML, to extend its use in access control systems. The WS-XACML standard describes formal mechanics to enable a mapping of P3P policies into XACML, which can be matched and evaluated using XACML PDP. In [149], the concept of XACML *AttributeSelectors* provides the technique for specifying constraints applicable to P3P where the *RequestContextPath* attribute can have a prefix value of "//P3P10/POLICES/", indicating that the policy instance refers to a P3P instance. Figure 7.12 is an example of the mapping between P3P and XACML. In this way, a communicating party can use P3P as the policy vocabulary for its *Capabilities,* to define PII that it is able and willing to release to a party, and can use P3P as the policy vocabulary for its *Requirements,* to describe its privacy preferences in terms of the usage of its released PII. This provides the mechanisms that allow both service provider and client to mutually negotiate for security and privacy preferences, using the capabilities of the W3C P3P policy framework already in place.

7.12 XTMAI Policy Framework

The XTMAI framework is a privacy trust authorization model infrastructure, developed by this work to intuitively and efficiently handle client-server privacy and confidentiality in a mutual manner. It empowers both client and server to perform security validation on each other before



```
<Apply FunctionId="urn:oasis:names:tc:xacml:2.0:function:xpath-expression-subset">

        <AttributeSelector RequestContextPath="//P3P10/POLICIES/POLICY/STATEMENT/PURPOSE/*"
        DataType="urn:oasis:names:tc:xacml.2.0:data-type:xpath-expression" />

        <Apply FunctionId="urn:oasis:names:tc:xacml:2.0:function:xpath-expression-bag">

                <AttributeValue DataType="urn:oasis.names:tc:xacml:2 0:data-type:xpath-
                expression">//P3P10/POLICIES/POLICY/STATEMENT/PURPOSE/current</AttributeValue

                <AttributeValueDataType="urn:oasis:names:tc:xacml.2.0:data-type:xpath-
                expression">//P3P10/POLICIES/POLICY/STATEMENT/PURPOSE/admin</AttributeValue>

                <AttributeValueDataType="urn.oasis:names:tc:xacml.2 0:data-type:xpath-
                expression">//P3P10/POLICIES/POLICY/STATEMENT/RECIPIENT/ours</AttributeValue>

</Apply>

</Apply>
```

Figure 7.12 WS-XACML - P3P Mapping

sharing sensitive information. To achieve this objective, the XTMAI framework makes use of two kinds of policies, namely:

- XACML policy rule: Purely for native XACML request decision evaluation;
- WS-XACML Assertion is an advertisement of policy assertions, usually to a remote party, which can be either a client or a server.

Even so, the two policy types require some kind of relationship between them, to link them up at runtime. Alternatively, the WS-XACML assertion policy can be constructed dynamically from the native XACML policy, but requires an efficient algorithm to do so. Figure 7.13 depicts a typical relationship between a XACMLPrivacyAssertion policy element and a primitive XACML policy. In this case, the XACML policy is strictly for traditional resource control and the WS-XACML is for privacy protection and negotiations. The *Apply* element, which is a XACML predicate, must be semantically correct in both policy rules. In typical usage, the service includes the *Apply* element in the *Requirements* section of its WS-XACML assertion, which the client must satisfy, and then the service can express what it is willing and able to do for a client in the *Capabilities* section of the WS-XACML policy.

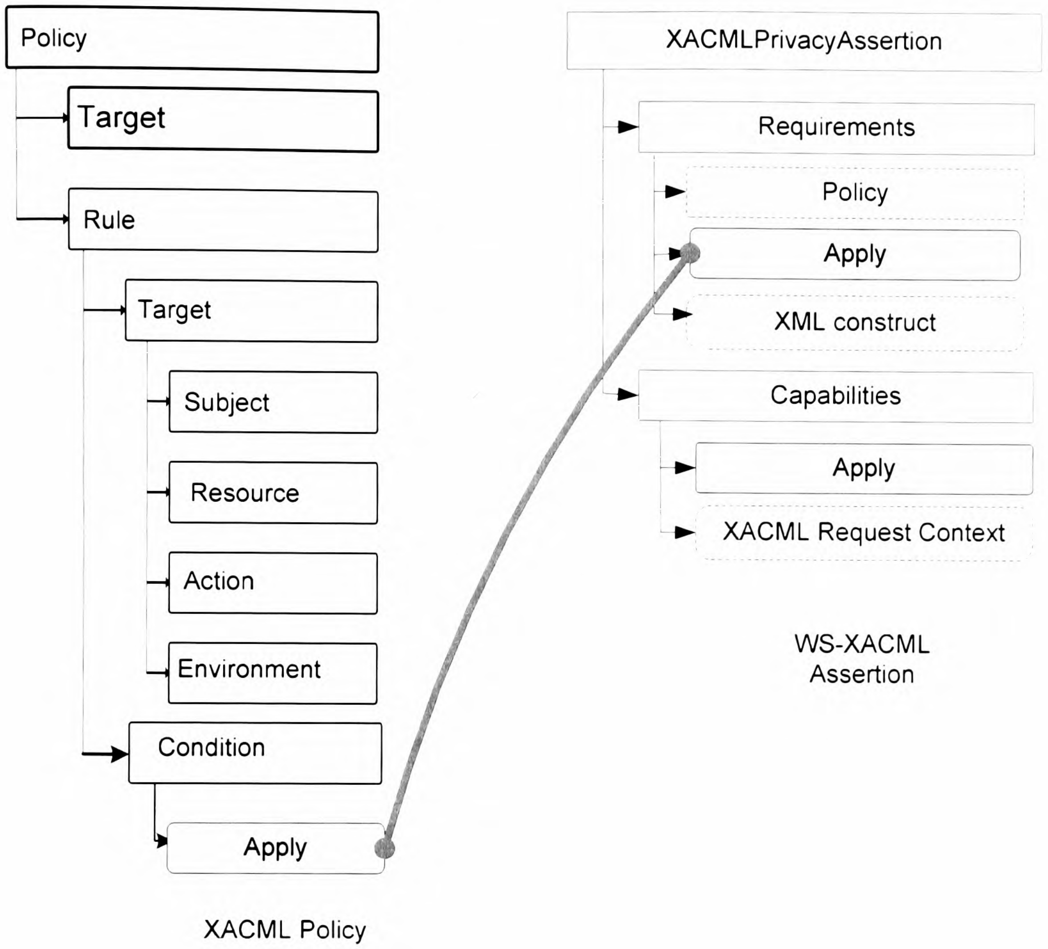


Figure 7.13 XACML Policy and WS-XACML Assertion Relationship

7.13 Conclusion

This chapter described the conceptual design and development of the access control framework. The analysis of the XACML actors in a distributed environment, are critical to the understanding of how to use XACML to address the privacy problem. Equally, the investigation and understanding of the trust models are instrumental in determining how trust relationships can improve privacy assurance in distributed environments. The threat modelling and the security requirement analysis supplemented earlier assumptions about privacy and confidentiality, and significantly influenced the conceptualization of the XTMAI infrastructure. Overall the rationale for choosing XACML and WS-XACML as policy candidates for the framework resulted from the various analyses of the requirements for addressing privacy and confidentiality problems considered.

Chapter 8. Obligation of Trust (OoT) Protocol

8.1 Introduction

So far, the general principles and conceptual framework that underline preserving privacy and confidentiality have been thoroughly examined and described, apart from remote enforcement of privacy. The basic concept is built upon the assumption that parties in distributed transactions have no means of enforcing obligating constraints placed on a remote party. In a traditional XACML model, an obligation is an action that should be performed by a PEP entity in conjunction with the enforcement of an access control decision. However, XACML describes an Obligation element as a set of attribute assignments, with an attribute FulFillOn which signifies whether the consuming PEP must fulfill the obligation if the access control decision is "Permit" or "Deny". When a PDP evaluates a policy containing obligations, it returns the access control decision and a set of obligations back to the PEP. However, in a distributed environment the service PEP is unlikely to be in the same security domain as the client. There is no guarantee that any obligations required by the client can either be incorporated into the policy used by the PDP, or even if they can, be enforced by the PEP. Given this, it makes sense to address the remote enforcement of obligations by allowing a service to convey back to the client an acceptance or rejection of their obligating constraints. The OoT protocol addresses this interaction.

In this chapter, the OoT protocol and its components are formally described. The encoding scheme and format describe the extension of a SAML request-response protocol schema, and how the WS-XACML assertion fits into the framework. Two trust negotiation strategies are described, which define the order and sequences of messages exchanged between negotiating parties plus the algorithm that implements the strategies. Additionally, the binding of the OoT protocol within the web services security environments in the SOAP header is described. Finally, the algorithms for matching two XACML assertions are presented.

8.1.1 The Formal OoT Protocol Definition

Obligation of Trust is a protocol that defines a standard mechanism enabling two or more communicating parties to exchange *obligating constraints* as well as *proof of acceptance*. The basic concept addresses the problem that a client currently has no means of enforcing obligations placed on a remote party. The protocol is divided into two steps: *Notification of Obligation* (NoB) (which may be signed or unsigned) and *Signed Acceptance of Obligation* (SAO) (which must be signed), and it is symmetrical. An initiating party sends a NoB outlining the obligating constraints it is placing on the other party, and the commitments it is willing to make if the other party accepts its obligations. The other party, after evaluation, sends back either a SAO of the constraints it accepts and the commitments it requires, or initiates more service negotiations with its own NoB, or rejects the request and terminates the session. Because the NoB and SAO are constructed using standard XACML policy constructs or XML documents, both communicating parties have a common language for expressing their requirements and commitments, and are able to feed these constraints directly into an appropriate decision engine and ensure their ultimate enforcement by their respective obligations services.

There are basically two ways the SAO can be constructed:

- The SAO can contain digitally signed *Requirements* and *Capabilities* of a party, signifying that this providing party is willing and able to provide the signed *Capabilities* if and only if the relying party satisfies the *Requirements.*
- Alternatively, the SAO can contain digitally signed *Capabilities* of the providing party and the *Capabilities* promised by the relying party. In this scheme, it indicates that the providing party agrees to release the *Capabilities* provided the relying party will reciprocate the same by releasing its own *Capabilities.*

The *Obligation of Trust* (OoT) defines a protocol for handling this important security aspect that allows the exchange of difficult-to-repudiate[13] information, and it is described in the following sections.

[13] The term "difficult-to-repudiate" rather than non-repudiation is used since repudiation is a legal issue that has to be determined in a court of law. The technical constructs in the context of OoT should make it more difficult for an entity to repudiate their actions.

Figure 8.1 is a sketch of the OoT protocol in operation, and shows how two parties may exchange signed components of the OoT. Party A wishes to access item X from party B, but it is assumed that party A knows nothing about the privacy or access control requirements for item X. Similarly, Party B knows nothing about the privacy requirements of Party A's attributes. Party A sends a request for item X and Party B responds with a NoB containing its *Requirements* and *Capabilities*. Figure 8.2 shows an outline of an algorithm for the decision making when a party receives a NoB. Party A checks whether it can satisfy Party B's *Requirements*, and whether party B's *Capabilities* can satisfy its own (party A's) *Requirements*. If Party B's *Capabilities* are acceptable and sufficient for Party A, and A can fully meet B's requirements, then A can send an SAO to B stating its pick of the offered *Capabilities* and its own *Capabilities* to meet party B's *Requirements*. If B's *Capabilities* are acceptable but not sufficient, or A has additional *Requirements*, A may send a counter NoB to B containing its additional or alternative *Requirements*. A's *Requirements* will determine the subset of B's *Capabilities* that it requires, and A may supplement them with additional ones of its own. A's *Capabilities* will include the subset of B's *Requirements* that it can provide, along

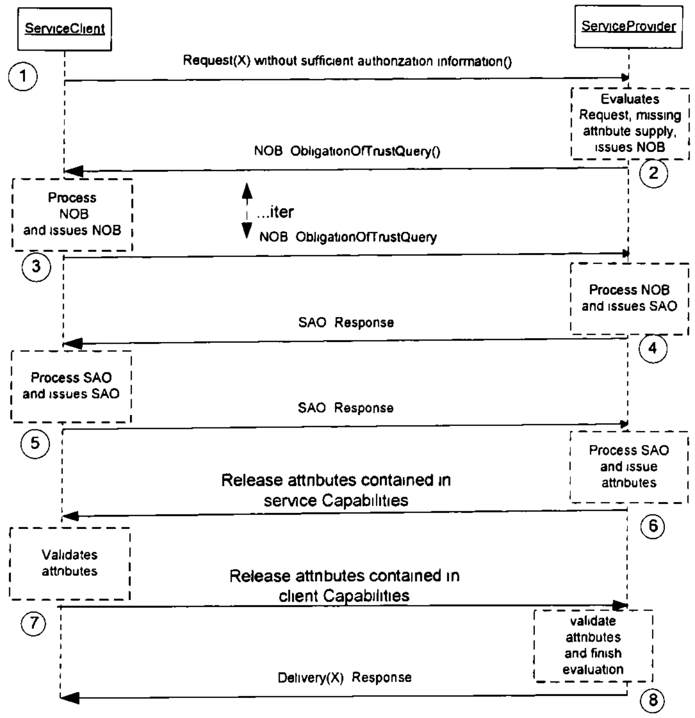

ServiceClient
ServiceProvider
1
Request(X) without sufficient authorization information()
Evaluates Request, missing attribute supply, issues NOB
NOB ObligationOfTrustQuery()
2
Process NOB and issues NOB
...iter
NOB ObligationOfTrustQuery
3
Process NOB and issues SAO
SAO Response
4
Process SAO and issues SAO
SAO Response
5
Process SAO and issue attributes
Release attributes contained in service Capabilities
6
Validates attributes
Release attributes contained in client Capabilities
7
validate attributes and finish evaluation
Delivery(X) Response
8

Figure 8.1 The OoT Protocol Sketch

with any additional ones it may be willing to provide. If Party B's *Capabilities* are insufficient for Party A, then A will either terminate the session or return a NoB with *Requirements* that supersede B's stated *Capabilities.* If A cannot meet all the stated requirements of B, then A may decide to terminate the session or add a reduced set of *Capabilities* to the NoB.

Party B evaluates party A's NoB and if satisfied with A's *Capabilities* and *Requirements* it returns a signed SAO stating in its *Capabilities* that it can fulfill all of party A's *Requirements,* and in its *Requirements* which of Party A's *Capabilities* it has chosen. If B is satisfied with A's *Capabilities* but not with A's *Requirements,* B may either send another NoB to A showing fewer *Capabilities* than A requires (along with its own *Requirements*), or terminate the session. If B is not satisfied with the *Capabilities* of A's NoB, it will either terminate the session or return a NoB with increased *Requirements.* If Party A receives another NoB, and this is satisfactory, it returns a signed SAO, otherwise it behaves as last time around. If Party A receives party B's SAO, and is satisfied with it, it returns its own signed SAO. Thus the parties continue to exchange NoBs until either party terminates the session (negotiated agreement is not possible) or returns a signed SAO. Once a signed SAO has been delivered, the recipient must either accept this by returning its own signed SAO or terminate the session. It is not allowed to return a NoB in response to a signed SAO, since this is in effect rejecting what one had previously offered in a prior protocol exchange. Once the negotiation is complete, and each party is in possession of the signed SAO of the other party, then Party A delivers the attribute values defined in *Requirement*s B, and Party B delivers item X to A.

```
Set flag initially to "SAO"
Evaluate received Requirements to determine whether I can meet them with my Capabilities
If so, construct offered Capabilities to match received Requirements
If not, either
terminate or
determine* whether additional Capabilities should be offered to match, and/or
construct Capabilities to match a subset of the received Requirements, plus additional alternative Capabilities to be offered, and set flag to "NOB"
Analyse Capabilities to be offered by me (as determined above) and construct a revised list of (my) Requirements.
Analyse sets of Capabilities received and compare with my list(s) of Requirements (as determined above).
If all my Requirements are met from one set of offered capabilities, keep the above-defined Requirements.
If all my Requirements are met from merged sets of offered Capabilities, construct Requirements from these, set flag to "NOB"
If my Requirements are not met, either
terminate or
determine* whether Requirements can be relaxed due to alternative Capabilities being offered and modify Requirements accordingly and set flag to "NOB"
If SAO flagged, send SAO, else send NOB.
(* "determine" could include the possibility to ask a human operator.)
```

Figure 8.2 Outline Algorithm for handling a NOB

In traditional trust negotiation, parties conduct bilateral negotiation based on challenge-response protocol in which in iteration, either a policy is released or a credential set to negotiate for service release. The OoT protocol reduces the amount of iteration in trust negotiation by combining the *Requirements* [14] (i.e. policies) and *Capabilities*[15] (i.e. credential descriptions) in iteration. The order and sequence of what a party is able to release is generally termed its negotiation strategy in the literature [118]. The section that follows describes two strategies.

[14] In the context of a traditional access control model, this is the policy or rules that govern access to protected resources.

[15] Similarly, in traditional access control, these are the attributes expected to satisfy the access control policy governing the requested resources. In an XACML context, this is called the Request Context.

8.1.2 OOT Negotiation Strategies

In traditional trust negotiation, a family of strategies determines the order and sequence of disclosure of policies and credentials. In [118, 162], a range of TN strategies are described to include eager, parsimonious, prudent, and hybrid strategies. The aim of using an interoperable strategy is to allow parties in negotiation to adopt a mutual order and sequence, which can make their negotiation, succeed whenever possible. In some cases, a negotiating party may not necessarily disclose credentials as demanded by a party, but instead, request that additional requirements be met by the party before disclosure of the requested credentials. The implication of not having a carefully crafted strategy is the possibility of recurring iterations. According to [163], properties of negotiation strategies include:

- a strategy which should advance in such a fashion that a deterministic outcome is reached, i.e. it should be complete;
- when success is impractical, failure should be gracefully communicated;
- a strategy which should optimize the iteration; justify the number of messages that have to flow in a typical negotiation session.

In the light of the above, descriptions of two strategies are given below along with their characteristics.

Parsimonious Strategy: Formally defined in [162], the parsimonious strategy characterizes trust requirement exchanges, which can advance to satisfying a specific trust goal. Theoretically, it kicks off by exchanging credential requests without the credentials, so that parties only exchange what has been unlocked by exploring all possible sequences of credentials that can be disclosed. The case here is slightly different because of the approach adopted. In this case, a party advertises its negotiation assertion that contains *Requirements* and *Capabilities,* which simplifies matters and makes the exchange more innovative. Recall that *Requirements* section contains constraints or rules the asserting party's recipient party must satisfy, and *Capabilities* that the asserting party is able and willing to do or release to another party if the requirements are satisfied. In this case, a party sends an SAO once a set of its *capabilities* can satisfy the asserting party's *requirements* on the one hand, and a set of the asserting party's *capabilities* can satisfy its local *requirements* on the other. In contrast, traditional trust negotiation sends either a credential or policy in iteration. However, the important thing to note about this strategy is that matching of the two alternate assertions must be TRUE, otherwise negotiation fails.

Persuasive Strategy: In some situations, either an asserting party's configured *capabilities* are insufficient to match a participant's *requirements*, or an asserting party's *requirements* are too great for a participant's *capabilities*. In this case the software might indicate to the recipient party that the participant's *requirements* are not covered by any of the asserting party's sets of *capabilities*. Theoretically, in this kind of situation, parties should be able to view the NoB request and possibly extend their *capabilities* or relax their *requirements* intuitively.

As an example, suppose a user has configured his *Requirements'* policy so that recipients are not to reveal the user's PII to third parties, but Service X offers very generous compensations or incentives (e.g. discount, 'buy one get one free', etc.) to Service C's users who are willing to sign up for X's new services. In this case, Service C could send the user a NoB containing a *Requirement* to provide permission for Service C to release their PIl to Service X, in exchange for compensation. The user's agent does not have a *Capability* to match this *Requirement*, so the user's client software could display Service C's *Requirement* for the granting of permission to forward the PII to Service X, along with Service C's *Capability* to offer compensation to the user. If the user dynamically chooses to accept this contract, a new *Capability* is added to the user's set of policy assertions, for this and future use, and a signed SAO is sent to Service C.

The strategy that addresses the use-cases is referred to as a *Persuasive Strategy*, which can allow parties to refine their *requirements* and *capabilities* intuitively, if the initial sets were partially unsuccessful. In this arrangement, a party checks if any set of its *capabilities* can satisfy participant's *requirements*, and checks if the other party's *capabilities* can satisfy its local *requirements*. If there are mismatches, the party can refine its own assertion, by perhaps offering the participant some sets of alternative *requirements* or *capabilities*, as the case applies, to advance the negotiation.

Analyzing the two strategies, it is obvious that the parsimonious strategy is a naive approach, which causes negotiation to fail once disclosed *Requirements* and *Capabilities* cannot match the other party's. In contrast, the persuasive strategy attempts cautiously to offer options capable of advancing the negotiation by proposing alternative *requirements* and *capabilities*. The benefit of the parsimonious strategy is its tight coupling to *minimal disclosure principles*; only the *capabilities* that can satisfy the participant's *requirements* are also included in the assertions and the minimal *requirements* that the other party's *capabilities* are likely to satisfy are included in the assertions. This scheme is suitable for simple cases where either party has no hidden requirements or rules that a party must unlock by a previous interaction.

It is important to mention that privacy trust negotiation, as described in this work, is a significant deviation from traditional TN, in the sense that a party discloses *Requirements* that contain constraints the participant must satisfy and *Capabilities* that contain what he is able and willing to do for the other party if its *Requirements* can be satisfied in a single iteration, which potentially reduces the number of iterations in a typical TN session. In sum, the concept of advertising a complete set of *Requirements* and *Capabilities,* by parties in business transactions, is plausible; it enables them to calculate the risk they may be exposed to before sharing sensitive information.

8.1.3 OoT Encoding Scheme

Using the built-in extensibility mechanism of WS-XACML and SAML Assertions, it is convenient to encode the components of the OoT as extensions of standard elements of a SAML request/response protocol. The NoB can be expressed as an instance of a *XACMLPrivacyAssertion* in which the desired obligating constraints are placed in the *Requirements* section of an assertion, and any obligations that the sender is willing and able to fulfill in the *Capabilities* section. The SAO can be expressed as an instance of a *XACMLPrivacyAssertion,* in which the *Requirements* section specifies the sender's understanding of what the recipient has committed to do, and the *Capabilities* section specifies the obligations that the sender has committed to undertake. In the privacy domain, these elements can be used to describe either the acceptable (*Requirements*) or supported (*Capabilities*) containing P3P policy constructs, also described in chapter 6. For example, if a recipient will only use the sender's sensitive information for the "current" transaction and "admin" purposes, and the information is only for the designated recipient, this can be sent as a P3P policy STATEMENT of PURPOSE expressed as a WS-XACML constraint. Figure 8.3 shows the basic structure of the OoT scheme; the schema is available at [164] and documented in appendix A. The schema defined new SAML *Request* and *Statement* message protocol types, which are described below:

ObligationOfTrustQuery: This container encapsulates one or more *XACMLPrivacyAssertions* that contain *Requirements* and *Capabilities* as a Notification of Obligation a party wishes to convey to another party in a privacy aware transactions.

ObligationOfTrustStatement: This container encapsulates a *XACMLPrivacyAssertion* that holds *Requirements* and *Capabilities* as a Signed Acceptance of Obligation a party wishes to convey to another party in privacy aware transactions.

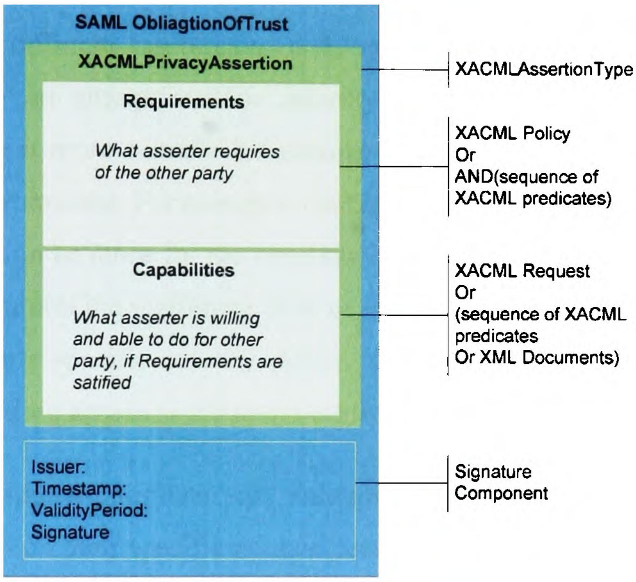


Figure 8.3 SAML Obligation of Trust Model

Both elements contain attributes the values of which describe some characteristics of the negotiation including:

- ootID: This attribute uniquely identifies a particular OoT context message, which can be referenced subsequently in the negotiation.
- InResponseTo: This attribute identifies whether the message communicated to a party is in response to a previous OoT context sent by the recipient party. For instance, an SAO response can be because of NoB or SAO context; since each message context is uniquely identified, it makes sense to let the communicating parties identify whether a message is a response or a fresh message context.
- IssuerInstant: This attribute relates to the instance at which the OoT context was issued.

The OoT protocol defined in the section above logically makes use of the set of assertions at different stages of the privacy negotiations. The WS-XACML privacy negotiation assertions can be constructed dynamically from the XACML policy or retrieved from a static location. When constructed dynamically, they will depend on the prevailing outcome of a previous negotiation state. Intuitively, the combination of XACML and SAML in the WS-XACML profile provides flexibility in the handling of an access control process that is complex in nature, whilst assuring respect for the communication party's privacy. This permits the combination of different autonomous policies into one applicable policy at runtime, which is fundamental to an enterprise-wide security policy administration. In particular, its core properties make it more suitable for simultaneous treatment of privacy and confidentiality in distributed environments. For example, the European Union [28] privacy directive requires that adequate provision be made for the free flow of information, meaning that an individual cannot use privacy to inhibit the legitimate flow of information. In this case, it is realistic to combine the organization's specific privacy policy, departmental privacy policy and user's privacy policy in controlling access to the user's attribute information.

Lastly, it is important to note that autonomous security domains control access to their resources, based on their specific domain security requirements, and administration. Thus, the combination of XACML, SAML and OoT provides the interface that seamlessly allows communicating participants to share attribute information in a more friendly privacy manner.

8.1.4 SAML OoT SOAP Binding Profile

The SOAP model provides "extensibility" points that allow other messaging protocols to be layered on top of it in a standard way. This convenient flexibility provides a rich mechanism for layering the OoT protocol with other existing security schemes. The Web Services security (WS-Security) is a set of specifications that describe the means for providing various types of security protection over SOAP payloads. Whilst WS-Security has defined SOAP profiles for authentication, data integrity and data confidentiality at the messaging layer, WS-XACML assertion aspects have not yet been addressed to cater for privacy as described in this work. There is a need to describe a standard way to use the WS-XACML profile to address mutual privacy and confidentiality in Web services scenarios.

On the basis of the above, a SOAP binding profile for the OoT exchanges is defined. This is expected to provide an appropriate mechanism to ensure that independently implemented compliant OoT systems can interoperate using standard messaging protocols. The existing <wsse:security> container in the SOAP header provides this natural way to carry the OoT message payload, as depicted in figure 8.4. One noteworthy benefit is that other existing security profiles can coexist with the OoT mechanism, which further supplements existing security services other than those addressed by the OoT technique. Shown in figure 8.4 is the complete logical structure of the OoT SOAP binding plus other individual components.

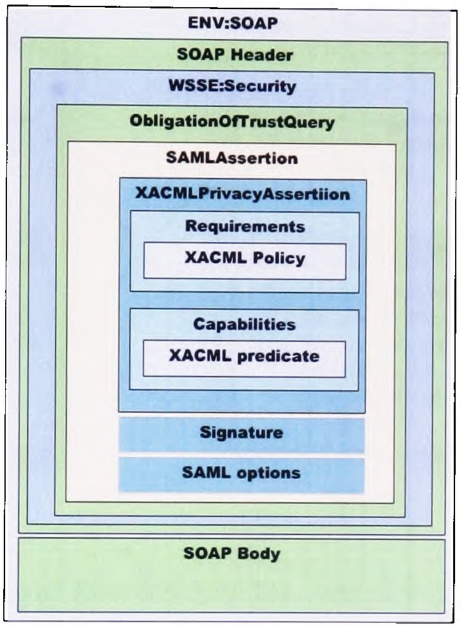


Figure 8.4 SAML OoT SOAP Binding Profile

8.2 XACMLAssertionType Matching and Evaluation

The comparison and matching of *XACMLAssertionType* are crucial aspects of the processing of privacy and confidentiality decisions. The purposes are to determine assertions' compatibility, and if so, which *Requirements* and *Capabilities* have common characteristics that can facilitate privacy negotiation and decision making. The matching phase is carried out by computing the intersection of the *Requirements* in foreign *XACMLAssertionType* with the *Capabilities* in the local *XACMLAssertionType*, which in effect can result in a new *XACMLAssertionType*. This new assertion may contain in its *Requirements* the interception of the original *Requirements* with the original *Capabilities* of the local *XACMLAssertionType*, and containing in its

Capabilities the intersection of the original *Capabilities* with the original *Requirements* of the foreign *XACMLAssertionType.*

Figure 8.5 illustrates the concept of matching two WS-XACML policies between a client and service in a typical negotiation interaction. Essentially, the outcome describes one or more instances of the requirements vocabulary that the owner of the assertion can expect from the other participating entity in the context of a policy *Target.* Similarly, the resulting *Capabilities* express one or more instances of the capabilities vocabulary, which the assertion owner can supply to the participating entity in the context of a policy *Target.* Thus an *XACMLAssertionType,* when evaluated as *TRUE,* indicates that the assertion is satisfied; and this is applicable, if and only if all the *Requirements* in the assertion evaluate as *TRUE* against an actual set of values for an instance of an associated policy target, otherwise it evaluates as *FALSE*, which indicates that the assertions were not satisfied. The rules and match algorithms for matching two compatible *XACMLAssertionTypes* are shown in appendix B and are adapted from [165].

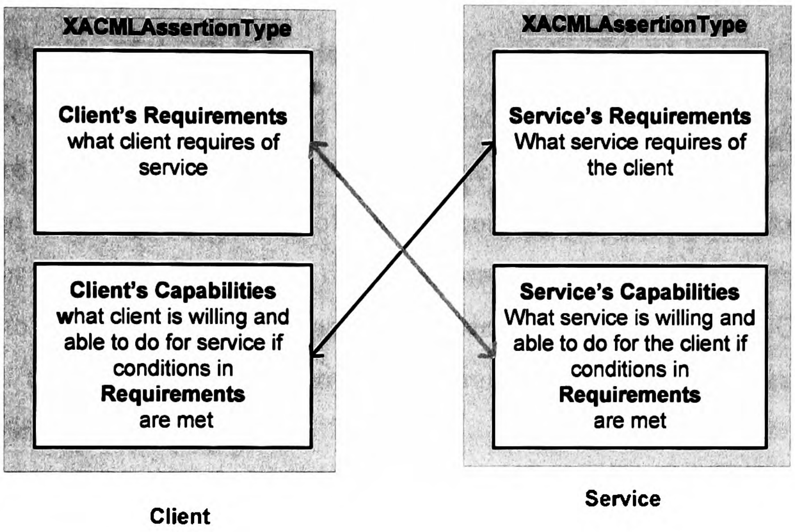

XACMLAssertionType
Client's Requirements what client requires of service
Client's Capabilities what client is willing and able to do for service if conditions in Requirements are met
Client
XACMLAssertionType
Service's Requirements What service requires of the client
Service's Capabilities What service is willing and able to do for the client if conditions in Requirements are met
Service

Figure 8.5 Matching of Two WS-XACML Assertion Types

8.3 The OoT Processing

The OoT negotiation message exchange can result in any of the four types of messages, namely: NoB encapsulated in *ObligationOfTrustQuery*, SAO contained in *ObligationOfTrustStatement*, and SAML AttributeStatement, also contained in

ObligationOfTrustStatement or in a Fault. Only one instance of the three types MUST be in the inbound or outbound OoT in a SOAP Header message context. According to the binding profile described in section 8.1.4, the OoT message is contained within the <wsse:Security> of the SOAP headers, and when included, MUST be processed by the recipient. The processing of the OoT context must conform to the required verification and validation rules defined in the SAML specification [153]. However, the processing of the OoT context further depends on the type of context.

8.3.1 Signing of the OoT Context

The rules applicable to signing the elements of <wsse:Security> capable of carrying a signature compliant with the XML signature standard within SOAP header block shall be observed when signing any part of the OoT context, particularly an SAO that MUST be signed [161]. In practice, it is expected that all message exchanges between two OoT parties shall be signed to ensure integrity, but particular attention is paid to the SAO, since it is devised to provide the mechanism for privacy assurance as a *proof of acceptance* of *obligating constraints*.

8.3.2 The Signing Key

There is usually a serious concern about the cryptographic key used in the production of a signature object; the signing key must be trusted by the relying party to belong to the subject presenting the SAO. Thus, the subject or the asserting party must be established by some mechanism, so that the relying party can be sure the entity is associated with the signing key. The PKI provides the mechanism that can establish or confirm that the asserting party possesses the signing key, but this requires that the parties must be PKI enabled, and obviously may require expensive processing of a trust path. The SAML standard defined two profiles for confirming a subject and can be used here.

Sender Vouches

In this profile, the attesting party, (presumed to be) distinct from the subject or negotiator, intends to vouch for the validity of the subject's attribute, such as the public key component of the subject. In order for the recipient to validate this claim, it must have an out-of-band existing trust relationship with the attesting party. The details of the Sender-vouches subject confirmation method are described in [161], with which both the attesting and receiver party must comply.

Holder-of-Key

In this profile, the attesting party, (presumed to be) the subject, includes an XML signature object that can be validated with the key information in the <saml:ConfirmationMethod> of the SAML assertion, and referenced for keyInfo by the Signature. Usually, the attesting party uses the holder-of-key confirmation method to express that it is acting as the subject of the SAML assertion containing the holder-of-key <saml:SubjectConfirmation> element. Again, the details of the holder-of-key subject confirmation method and processing are defined in [161]. In this scheme, since there is no out-of-band trust relationship, the subject must show sufficient knowledge of the confirmation key, by using the confirmation key to sign content within the message context, and the results must be included in <ds:Signature> element as specified in [161].

8.3.3 OoT Security Timestamps

The OoT message container contains an IssueInstant attribute which is employed to express the creation time of the security semantics in an OoT message context. What it does not contain is the expiration time requiring this to be done by application specific runtime policies. An alternative approach is to make use of the <wsu>Timestamp> element described in [161] to provide the timestamp service.

8.3.4 Processing of Notification of Obligation

According to the OoT schema definition, the *ObligationOfTrustQuery* instance can contain one or more instances of *XACMLPrivacyAssertion* that hold *Requirements(R)* and *Capabilities (C)*. Similarly, the local *XACMLPrivacyAssertion* must have the same semantic structure and vocabulary. To differentiate the two assertions, the following notations apply.

- Let R_F = *Foreign Requirements* contained in the incoming NoB assertion;
- C_F = *Foreign Capabilities* contained in the incoming assertion;
- R_L = *Local Requirements* contained in the local assertion;
- C_L = *Local Capabilities* contained in the local assertion.

Figure 8.6 shows a NoB decision table detailing the possible combinations and the outcomes. In processing a NoB context, a check MUST be done to determine whether the context is in response to a previous NoB, or a fresh NoB. This is done by checking the InResponseTo attribute; which when empty means that the particular NoB has no previous context, and as such must be treated as a first NoB. Whereas, if the InResponseTo attribute is not empty, it means that the NoB is in response to an NoB previously sent by the recipient and is identified by the InResponseTo attribute value. The party receiving this MUST check the value against its session management mechanism that it generated in the earlier NoB, to which the reference is made.

$C_L >= R_F$		Y	Y	Y	N	N	N	N	N
$C_F >= R_L$		Y	U	N	Y	Y	N	N	N
Can construct C_L to match R_F					Y	N	Y		N
Provider can construct C_F to match R_L			Y	N			Y	N	
Indeterminate	Y								
Terminate						X		X	X
Send SAO		X							
Send NoB			X						
Send NoB with new Req or Cap					X		X		
Send indeterminate				X					
Human intervention	X								

Y = yes, N= no, U= indeterminate, X=outcome

Figure 8.6 NOB Decision Table

However, in negotiation, two negotiation parties must adopt the same negotiation strategy described above for their negotiation to succeed or fail gracefully.

Note the SAO is sent once a party is happy with the sets of *Requirements* and *Capabilities* and doesn't wish to send another NoB. Once this is the case, the party MUST sign the SAO, and it expects the recipient party to reciprocate with its SAO.

8.3.5 Processing of Signed Acceptance of Obligation

The SAO context is a signed OoT context that contains one instance of XACMLPrivacyAssertion which MUST be verified and validated to ensure it has not been tampered with or arrived corrupted, and then logged to preserve the acceptance as too-difficult-to-repudiate technical evidence in the case of privacy breaches. The generation and processing of SAO must follow the same rules described in[161] for processing an XML digital signature compliant. Figure 8.7 depicts a SAO decision table. There are two cases in which to generate a SAO context; either when a foreign assertion is compatible with a local assertion [S(M) = C(M)] or a fresh SAO is received in response to the processing party's previous SAO. In which case the table shows the message that MUST be in the OoT response context.

S(M)= C(M) Received SAO	Y Y	Y N	N Y	N N
Send SAO		X		
Send unlocked Attributes	X			
Send indeterminate			X	X

Y = yes, N= no, X=outcome

Figure 8.7 SAO Decision Table

8.3.6 Processing of AttributeStatement

The processing of the OoT AttributeStatement MUST adhere to the processing rules specified in [153] in the first place, and subsequently the attribute values should be passed back to the component that consumes the attribute information. Usually, the attribute information shall be passed to XACML PDP via its ContextHandler to enable it to complete the access control decision.

8.3.7 Processing of a Fault

Standard SOAP error conventions MUST be taken into account. Errors as a result of OoT processing MUST be handled using a special container <NegotiationState>, and included in the SOAP body.

8.4 Conclusion

This chapter described the core concepts underlying the OoT protocol, WS-XACML construct, and matching and evaluation of alternate privacy assertions. Remote enforcement of privacy obligations is a critical requirement for privacy assurance, expected to enhance communicating parties' confidence that their sensitive information will be treated in a manner that respects their privacy. Although XACML defined a standard *Obligation* element with a set of attribute assignments, so that a consuming PEP can fulfill the obligations in conjunction with the enforcement of access control decision, the obligations do not actually reflect the client's privacy obligations.

The concept of the OoT model addresses this problem by enabling both the service and client to convey privacy obligating constraints and an acceptance or rejection of the constraints in a standard way. The privacy assertion encoding scheme, SOAP Binding profile for the OoT context, and sketch of the algorithms for processing the OoT message context have been provided. Although the OoT protocol is a significant development in addressing the remote enforcement of privacy and confidentiality guarantees, the formal verification of the protocol and algorithms would require a separate study and has been suggested for future work.

Chapter 9. Implementation of an XACML Trust Management Infrastructure (XTMAI)

9.1 Introduction

Henver et al [52] stated that "design research addresses research through the building and evaluation of artefacts designed to meet the identified business need". So far, the underlying conceptual elements underpinning this work have been described in the previous chapters, particularly chapters 7 and 8. The purpose of this chapter is to describe the software implementation of the resultant technical framework in Java programming language as a proof of concept. In practical terms, the resulting software is expected to address the privacy and confidentiality concerns already identified in the previous chapters. That is, the sharing of information between access control systems without compromising the confidentiality and privacy of personal information. The software design rely on object-oriented(OO) programming features, and it is divided into packages and classes using proven OO design strategies [166, 167], such as. *AbstractBuilder, AbstractFactory* patterns, etc. The XTMAI model makes use of two types of policies:

- Native XACML policy based on XACML version 2.0;
- XACMLPrivacyAssertion based on WS-XACML.

The policy framework that defines the rules governing the protection of privacy and confidentiality is structurally complex; as such, it requires a careful design strategy to abstract the complex operations, and the encapsulation of the various functionalities. This is implemented using the *builder* design pattern [166], and a UML class diagram that models the participating components. It is essential to mention here that whilst the native XACML policy can be made static, XACMLPrivacyAssertion will most frequently be constructed dynamically based on the underlying privacy negotiation requirements. Additionally, XACMLPrivacyAssertion can be constructed from pure XML constructs or derived from a P3P policy mapping to XACML at runtime.

This chapter focuses on the implementation of the software system supporting the XTMAI model, describes the software architecture, and the core APIs, especially the SunXACML [168], which provided many of the access control functionalities. In addition, the various UML class diagrams of the core XTMAI modules are then presented, with a description of their basic features. Lastly, how XTMAI works is described as well as the technical validation of it.

9.2 Overview of System Infrastructure

Figure 9.1 depicts the XTMAI stack showing the core supporting APIs. Since the infrastructure is symmetrical, it requires that communication participants i.e. party A and B in a privacy aware environment should have identical software at both communication endpoints. The core supporting APIs provided the abstract functionalities upon which specific business logic was constructed. In particular, the SunXACML API package is significant and central to the software implementation.

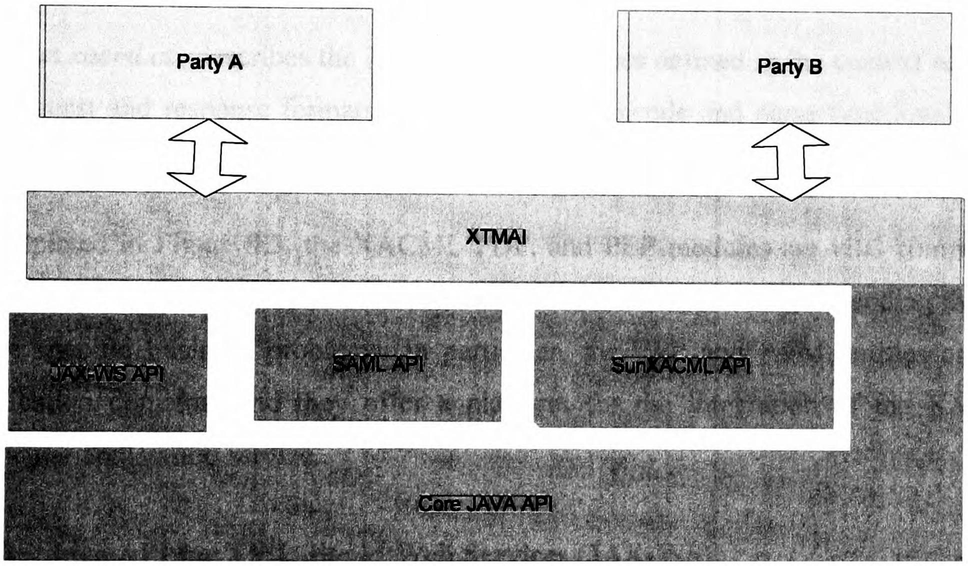


Figure 9.1The XTMAI Framework Stack

9.2.1 SunXACML API.

The SunXACML API is a well-documented open source implementation of the XACML standard, which provides an extensive suite of Libraries to facilitate XACML compliant application development, and it helps developers to concentrate on business logic concerning access control operations. The following section overviews the main components of the API.

com.sun.xacml is the key package, which contains the core XACML components and logic for matching and evaluating policy and PolicySet. In addition, it contains the PDP class used for the processing of access control decisions.

com.sun.xacml.attr supports all the standard XACML attribute data types including designators, selectors, and the necessary base *Factory* objects to allow for the creation of new attribute types and values.

com.sun.xacml.combine is the package that contains all the standard combining algorithms, interfaces for building new ones, and the base *Factory* objects for extending the algorithms.

com.sun.xacml.cond is the package that provides standard conditions and functional logic. It also includes standard interfaces and classes for building new functions.

com.sun.xacml.funder offers support for locating attributes required by the PDP, such as finding applicable policies, looking for attributes not supplied by the request, resolving resource identifiers, as well as a set of classes that extends the usage of this package.

com.sun.xacml.ctx describes the XACML context types defined in the context schema, such as the request and response formats. It provides both encode and parse functionalities to enable XACML components to communicate with one another.

As depicted in Figure 9.3, the XACML PDP, and PEP modules are vital components of the architecture. The SunXACML API provides an extensible point for extending the modules to solve specific business problems. In particular, the PEP and AFM entities are specific to application domains; and they offer a platform for the integration of the XACML into a particular application service.

9.2.2 Java API for XML-Based Web Services (JAX-WS)

JAX-WS is an extensive pluggable framework, and an important segment of the Java EE 5 platform [169]. The Java EE is designed essentially for implementing enterprise-class SOA and emerging web technologies. JAX-WS provides the mechanics for writing XML-based web services, i.e. SOAP messages aimed at simplifying the task of developing web-base software using Java technology. It provides support for multiple protocols such as SOAP 1.1, SOAP 1.2, and other XML constructs. This API is used here for service layer integration, as well as handling the negotiation protocol based on SOAP headers capabilities. Readers interested in this API should visit the community website [170]. Figure 9.2 depicts the JAX-WS Service-Client architecture in the context of a SOAP message exchange between a client and a service.

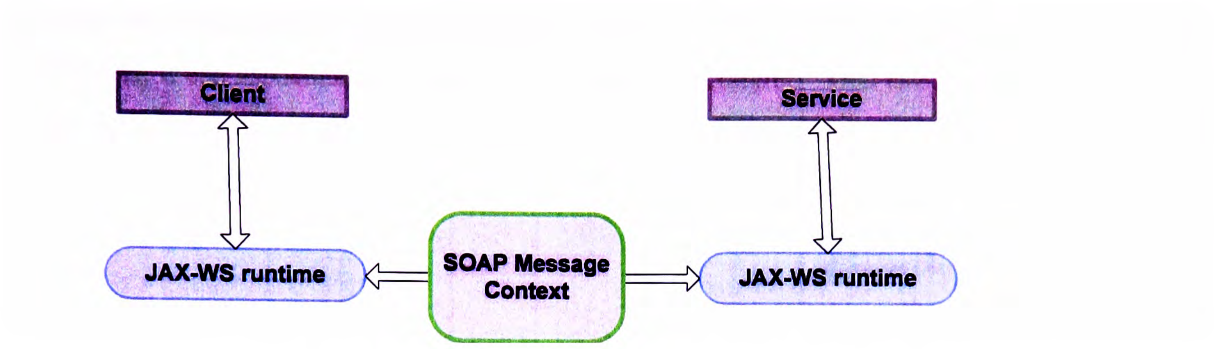


Figure 9.2 JAX-WS Service-Client Architecture

9.2.3 The SAML API

The SAML API contains base libraries that support the implementation of the SAML assertion related security services used by the infrastructure. Here, the SAML API is provided by Sun's Access Manager SAML API release. The API supports most of the SAML version 2 specifications, but it is used in this work for the signing and validation of assertions, since the request-response message exchange is implemented as part of the OoT protocols.

9.3 The XTMAI Architecture

Figure 9.3 shows the core XTMAI architecture, and depicts two modes of interactions: service request/response and trust negotiation (TN) modes. The service request mode handles service request/response messages, and can support any Internet application Protocol or standalone clients. The TN mode is a backchannel that allows both XTMAI compliant systems to engage in privacy negotiation that will result in final access control decisions. To protect privacy, it is assumed that the initial service request context will contain a provable token obtained during the authentication phase that describes among others, the *subject* attributes in a native XACML request context, which a XACML PEP can pass to the PDP. It is important to mention that the token (usually an opaque handle) is critical to the trust negotiation phase between the two XTMAI compliant systems. It provides the initial mutual authentication, which establishes the trust context upon which subsequent high-level trust is built. In order to use the handle in the XACML request context, it should be mapped into the *subject* attribute of the XACML request context. The significant thing is that this initial token should be linked to the initiator's attribute information by its authentication service, and passed to the initiator's XTMAI so that when the other privacy negotiating party sends an OoT message context to the initiator's XTMAI, it can associate the request with a particular initiator. This establishes the basis to use the initiator's privacy assertions to negotiate with the other party.

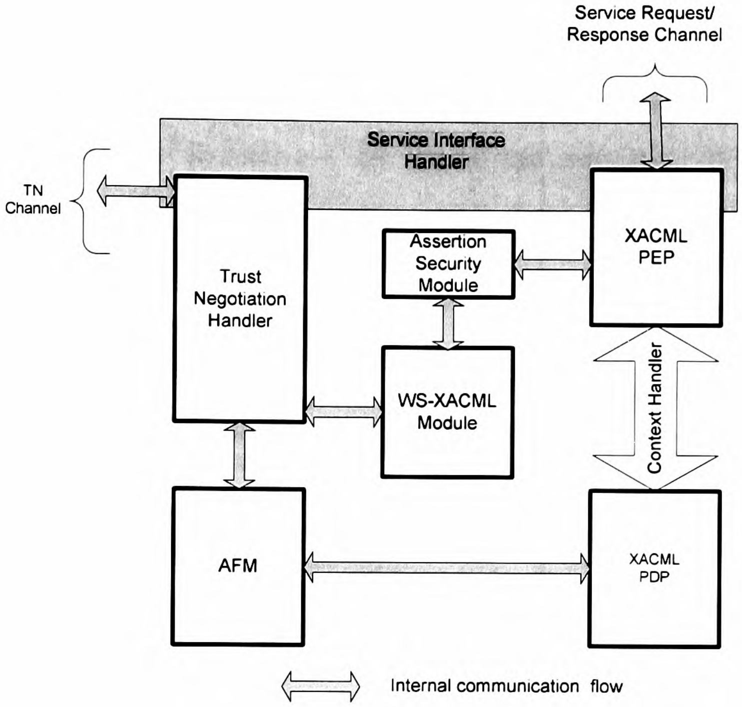

Service Request/
Response Channel
Service Interface Handler
TN Channel
Trust Negotiation Handler
Assertion Security Module
XACML PEP
WS-XACML Module
Context Handler
AFM
XACML PDP
Internal communication flow

Figure 9.3 The XTMAI Architecture

9.3.1 Service Interface Handler (SIH)

The Service Interface Handler (SIH) is an application specific security handler, which works together with the TNH and XACML PEP modules. Its purpose is to support other application layer protocols such as HTTP, FTP; SMTP, etc in a pluggable manner. In this implementation, the JAX-WS handler framework [170] [171] are utilized for pre-processing of inbound and outbound SOAP messages, such as security information contained in SOAP message headers. From the viewpoint of the SOAP message processing, the idea is to process security related information before processing other parts of the request by the endpoint. In JAX-WS API, two types of handlers: protocol (or SOAP) and logical are defined. The difference between the two is that protocol handlers are specific to a protocol, and can manipulate any part of the message, i.e. message header, whilst logical handlers are protocol-agnostic, they can modify only the body part of the message context. Figure 9.4 depicts the handlers and their logical workflow.

and illustrates how both handlers can access and modify a part of a message context. The diagram demonstrates the coherent processing of the handlers in the context of service-client interactions. From the client's perspective, in an outbound message context, the logical handlers are processed before the protocol handlers. Conversely, in the inbound mode, the protocol handlers are processed before the logical handlers. The case is similar from the service's perspective. Figure 9.5 depicts the part each handler can access and modify in the SOAP message context. The protocol handler is specifically used in the implementation of the PEP and the TNH modules respectively, to handle the processing of security related information within the SOAP Header container in order to determine whether access can be allowed to a protected target.

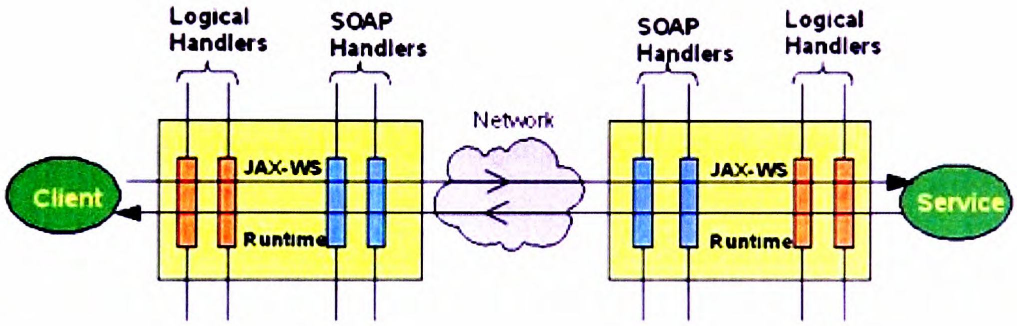


Figure 9.4 SOAP handlers Interaction in Service -Client Request- Response [170]

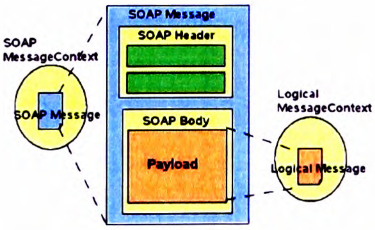


Figure 9.5 SOAP Message Context [170]

9.3.2 XACML PEP

The PEP is an application specific component that intercepts access requests, where appropriate, asks the PDP for an access control decision based on the attribute information made available to it, and then enforces the decision passed back by the PDP. In many access control paradigms, the PEP serves as a gatekeeper to the protected resources or targets. This implies that the access filtering logic and session management be implemented at the PEP as described in chapter 7. This requires intercepting all incoming requests directed to the protected *Target*, which should also determine whether access control is required[16]. The PEP interacts with the PDP based on the Context Handler, which encodes the request-response contexts in a manner that both can understand. Figure 9.6 shows the UML Class diagram and classes that participate in the XACML PEP Module. The following section briefly describes the object classes.

MessageContext Class
This is a core JAX-WS API class which abstracts the SOAP message context, and provides convenient abstract methods for the processing of a SOAP message context.

XTMAIBaseSOAPHandler Class
This is a base class that extends *MessageContext* class, and defines a type-safe attributes for the message context. The core methods defined in this class are briefly described as follows:

- The *initHandler()* method, which has @postConstruct annotation and it is invoked to initialize a set of parameters just prior to calling any message context processing method.

- The *destroy()* method that is called for the destruction of handler instances, which cleans up computing resources used during the handler operation.

- The *handleFault()* method that is invoked for fault message processing.

- The *close()* method that is invoked at the completion of a message exchange pattern before the JAX-WS runtime dispatches a message, a fault or an exception.

[16] This applies in the case where the access control decision state is kept for sometimes to avoid asking an already authorized user to resubmit context decision information.

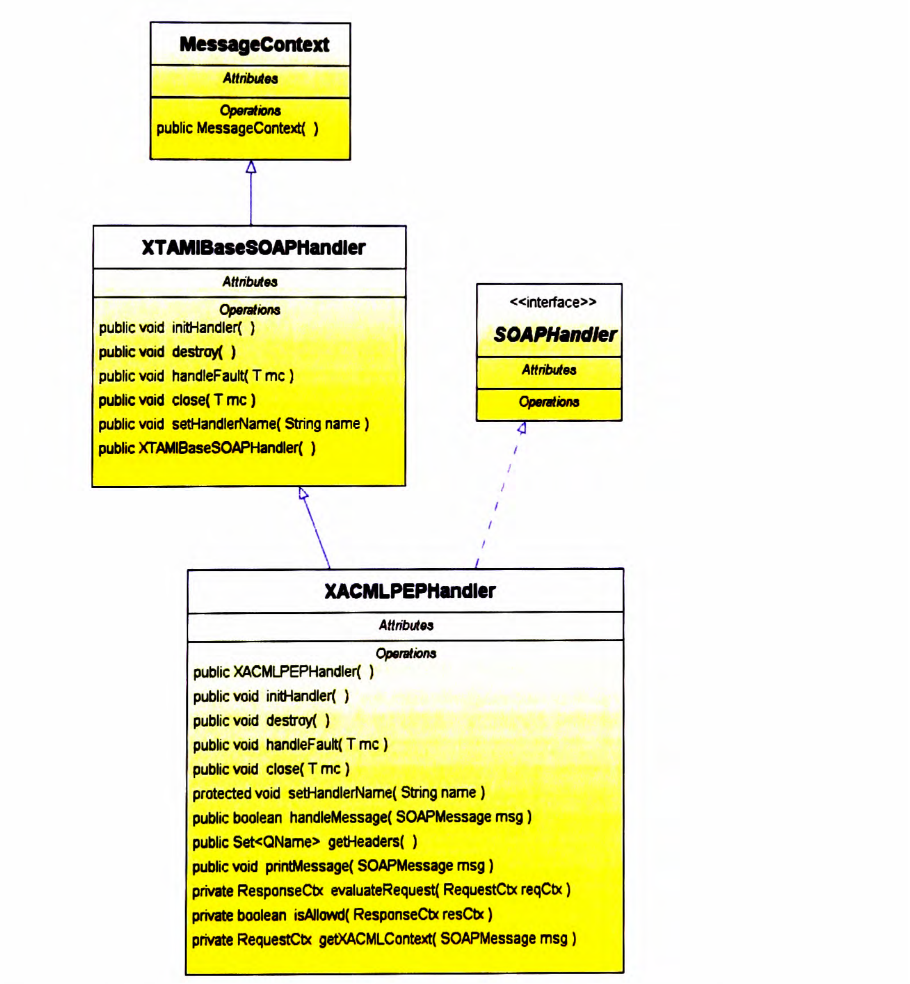
- The *setHandlerName()* method that is called to set the name of a specific handler service.

SOAPHandler Class

The *SOAPHandler* is a core JAX-WS class, which defines a type-safe attributes for protocol handlers, and provides convenient mechanisms to operate on the SOAP message for either the *request* or *response* context. The methods defined in this class provide the functionalities that are used to operate on the SOAP header by the concrete handler service implementation.

XACMLPEPHandler

The *XACMLPEPHandler* is the concrete implementation of the handler service. It extends the features of the *XTMAIBaseSOAPHandler* and *SOAPHandler* classes, and implements methods



MessageContext
Attributes
Operations
public MessageContext( )
XTAMIBaseSOAPHandler
Attributes
Operations
public void initHandler( )
public void destroy( )
public void handleFault(T mc)
public void close(T mc)
public void setHandlerName(String name)
public XTAMIBaseSOAPHandler( )
<<interface>>
SOAPHandler
Attributes
Operations
XACMLPEPHandler
Attributes
Operations
public XACMLPEPHandler( )
public void initHandler( )
public void destroy( )
public void handleFault(T mc)
public void close(T mc)
protected void setHandlerName(String name)
public boolean handleMessage(SOAPMessage msg)
public Set<QName> getHeaders( )
public void printMessage(SOAPMessage msg)
private ResponseCtx evaluateRequest(RequestCtx reqCtx)
private boolean isAllowd(ResponseCtx resCtx)
private RequestCtx getXACMLContext(SOAPMessage msg)

Figure 9.6 XACML PEP Module UML Class Diagram

that are used in message context processing. Many of the methods defined in this class are already described above. The two important methods that are central to the processing of the message context are described below.

- The *handleMessage() is a* method that handles the SOAP Message Context, and using the in-built MESSAGE_OUTBOUND_PROPERTY, it can determine whether a message context is inbound or outbound, and then processes the message appropriately based on the context type. In processing a message context, it invokes the *evaluateRequest()* method, passing the XACML *request context* object, which is extracted from the inbound message context contained in the SOAP header container.

- The *evaluateRequest()* is a method that processes the XACML *request context* by sending a decision request to the XACML PDP module, which returns a XACML decision *response context* object. The *response context* object is then passed to the *isAllowed()* method, which interprets the decision response to determine whether the PEP should allow access or not.

9.3.3 XACML PDP

The XACML PDP is the access control decision engine that makes its decision based on the local policy against the *request context* passed onto it. One of the core features of the PDP is the ability to find the correct policies and attributes that apply to a specific request. The PDP typically uses the information contained in the *request context* to locate the appropriate access control decision information to use. In some cases, some of the *request context* attributes are missing; then, the PDP naturally invokes one or more PIPs in order to locate and retrieve the missing attributes. This particular feature is what makes XACML ideal for privacy negotiations. In this implementation, the Attribute Finder Module (AFM) provides the interface between the XACML PDP and PIP components. The PIP in this case may be an AA or IdP/STS where the client is originally authenticated before web service invocation. The XACML PDP module in this implementation is an extension of the SunXACML PDP engine.

9.3.4 Attribute Finder Module (AFM)

The SunXACML API provides suitable mechanisms that help the PDP to find missing context information (attributes) during policy evaluation. The concept of the Attribute Finder Module (AFM) is one such convenient service, which aids in the location of missing context information. The PDP calls this service through the Context Handler whenever it cannot find sufficient information to process a request. Usually, either the missing information is defined in the *Rule Target* or *Condition* instance; then, using the notion of *AttributeDesignator*, described in the local policy, search is performed through the runtime Finder Module instances configured with the XACML PDP engine during setup. Doing this requires that sufficient information be passed to the AFM module to enable it know where to ask for the missing attributes. These features are critical to the launching of the privacy negotiations using the combined OoT protocol discussed in chapter 8. Figure 9.7 shows the AFM UML class diagram and the participating object classes. The following section describes the main classes.

AFMWSClient Class

The *AFMWSClient* is an *interface* that abstracts the JAX-WS client instantiation, which provides support for the dynamic invocation of another XTMAI service endpoint.

AttributeFinderModule Class

The *AttributeFinderModule* is an abstract base class of the SunXACML API, which provides a support for locating and retrieving of missing information in XACML request context.

AttributeFinderModuleImp Class

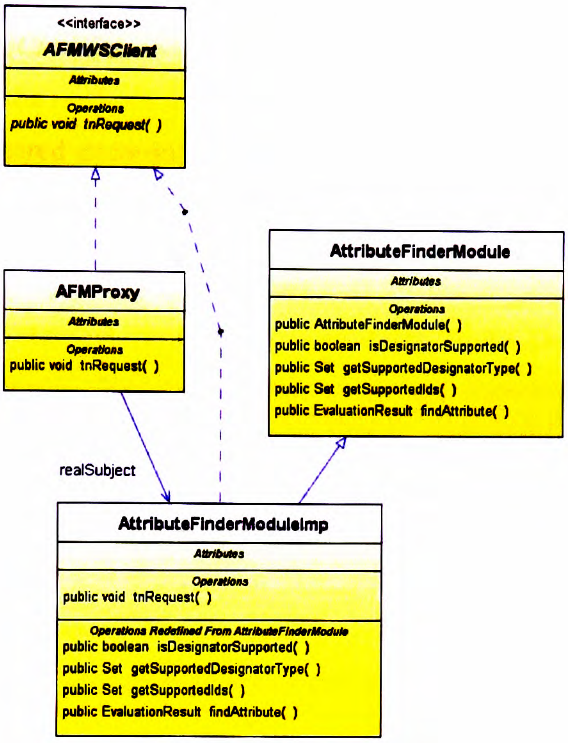
The *AttributeFinderModuleImp* class is derived from the *AttributeFinderModule* class, and implements the methods that provide the context attribute finding features. The implemented methods are those defined in SunXACML AFM class.

AFMProxy Class

The *AFMProxy* is the web service client implementation, which initiates interaction with another XTMAI aware service by invoking the TNH module. The TNH handles the actual negotiations between communicating parties, and returns a response to the *AFMProxy*, which is then passed back to the XACML PDP to complete the access control decision process.

9.3.5 Trust Negotiation Handler (TNH) Module

The TNH manages the ordering and sequencing of privacy negotiations during OoT protocol message exchange. It intercepts an inbound message, determines the type of OoT message context and performs the required processing. This module also sets the outbound OoT response message context based on the outcome (OoT assertion matching result) of the processed inbound message context. The class diagram of the TNH module is documented in appendix D, and shows the classes that participate in the TNH module. The following section describes the object classes.



<<interface>>
AFMWSClient
Attributes
Operations
public void tnRequest( )
AFMProxy
Attributes
Operations
public void tnRequest( )
AttributeFinderModule
Attributes
Operations
public AttributeFinderModule( )
public boolean isDesignatorSupported( )
public Set getSupportedDesignatorType( )
public Set getSupportedIds( )
public EvaluationResult findAttribute( )
realSubject
AttributeFinderModuleImp
Attributes
Operations
public void tnRequest( )
Operations Redefined From AttributeFinderModule
public boolean isDesignatorSupported( )
public Set getSupportedDesignatorType( )
public Set getSupportedIds( )
public EvaluationResult findAttribute( )

Figure 9.7 AFM UML Class Diagram

XTMAIBaseSOAPHandler Class

This class and its methods are described in section 9.3.2

SAMLOoTNegotiationMessageHandler Class

This class implements the SOAPHandler class and extends the XTMAIBaseSOAPHandler class to provide concrete logic for processing the message context. It implements the *handleMessage()* method which is also described in section 9.3.2. In a typical process, if the message context is inbound, it calls the SAMLOOTMessageCtx object passing a MessageContext object as a parameter, whereas for outbound, it retrieves the *response context* message that is set by the *setContextResponse()* method and binds it to the SOAP Message Header for outward transmission.

SAMLOoTMessageCtx Class

This class defines the methods that handle the inbound message context. The inbound message context is first filtered according to the OoT message type, and then passed to one of the context processors described below to handle the actual processing of the message context.

WSXACMLContextProcessorFactory Class

This is an abstract class which defines two methods for the OoT message context processing. Each context processor class extends this class, and implements the methods to provide a specific business logic functionality. The class defines the following abstract methods:

- The *doProcessMessage()* method that handles the processing of the OoT message context, and this depends on what message type it receives. There are four types of such messages, namely *ObligationOfTrustQuery*, *ObligationOfTrustStatement, SAMLAttributeStatement*, or a *Fault* message.

- The *setContextResponse()* method that handles the setting of an appropriate *response context* message after the *doProcessMessage()* method operation is completed.

WSXACMLNOBContextProcessor Class

This class extends WSXACMLContextProcessorFactory, and provides the business logic (implemented on the *doProcessMessage()* method) for the processing of the Notification of Obligation (NoB) message context. The NoB context is encapsulated in an ObligationOfTrustQuery element of the OoT message context. It is important to mention that the processing of a NoB context can take significant computing resources, i.e. the matching and evaluations of two *XACMLPrivacyAssertions,* which uses one of the strategy algorithms discussed in chapter 8. The local XACMLPrivacyAssertion containing *Requirements* and *Capabilities* is matched against the received *XACMLPrivacyAssertion*, which also contains *Requirements* and *Capabilities*. The matching result determines the type of outbound message context to generate, and it is set using the *setContextResponse()* method.

WSXACMLSAOContextProcessor Class

This class is invoked when the received message context is a Signed Acceptance of Obligation (SAO). In processing the SAO context, it invokes the Assertion Security Module objects to perform security functions such as verification and validation of the SAO, and then using the *setContextResponse()* method, it sets an appropriate outbound message context. It is equally important to mention that the task handled by this class involves some cryptographic operations; which can take large amount of computing resources.

WSXACMLAttributeContextProcessor Class

This class is invoked if the received message context is a SAML AttributeStatament. In this case, if it is a signed assertion, security functions are performed, using the Security Assertion Module to verify and validate the assertion, before it is passed to the consuming object. In most cases, the attributes are passed to the AFMProxy which initiated the trust negotiation, and then to the PDP. Usually, this process completes the trust negotiations, and depending on the evaluation result, the PDP sends an appropriate XACML *response context* to the PEP for enforcement.

SAMLOOTFaultContextProcessor Class

This class is invoked when the received message context contains a fault due to OoT processing. There may be cases when the match operations can result in *Indeterminate* as defined in the XACML standard. In this case, a response message can be sent to a party with an indication of what caused the negotiation to fail. Ordinarily, this is not a SOAP fault, and is treated separately from a normal SOAP context fault. The SOAP context fault is handled by *handleFault()* defined in SOAPHandler implementations.

9.3.6 Assertion Security Module (ASM)

The Assertion Security module handles all the security related operations. This module uses the SAML API provided by Sun's Access Manager SAML API release and has two classes, one for performing the signature scheme and the other for verifying and checking the validity of signed objects.

9.3.7 The WS-XACML Module

The WS-XACML Module defines the auxiliary classes used by the TNH module for encoding, parsing and matching of policy assertions based on the WS-XACML match algorithms. The module contains the classes that are used for the construction of the WS-XACML and OoT elements. Because the construction of XACMLPrivacyAssertion components is complex, the *Builder Design Pattern* [166] is used to abstract the essential details of the underlying complex objects. Figure 9.8 depicts the UML class diagram, and the components that participate in the WS-XACML module. The following section briefly describes the main classes.

ApplySection Class
This class provides a convenient representation of the Apply element, which can contain one or more *AttributeValue*, *AttributeDesignator* and *AttributeSelector* elements. These elements have their respective classes, which typically define the *accessor* and *mutator* methods for their respective attributes.

Apply Class
This class provides the methods for creating the *Apply* sub elements using the *ApplySection* class and its methods.

Policy Class
This class is used to construct the XACML native policy. The SunXACML API provides *helper* classes for doing this. In addition, static policies can be loaded through this class, and then added to the XACMLPrivacyAssertion policy.

RequestCtx Class
This class is used to construct a native XACML *request Context*, which can be added to the *Capabilities* element of the *XACMLPrivacyAssertion* element, or used by a PEP to construct a *request context* object that is presented to a PDP.

Vocabulary Class
This class constructs a *Vocabulary* instance, as defined in the WS-XACML specifications. This implementation supports three vocabulary types, namely xacml, p3p and xml. The *Vocabulary* instance is a critical factor in the policy assertion matching process. For two XACMLPrivacyAssertion policies to be matched, their *Vocabulary* instances must be compatible, otherwise the matching of other segments of the assertion must be aborted, i.e. the evaluation of the assertions must not proceed. The basic idea is to ensure that the semantics of the assertions domain are applied correctly during the matching and evaluation.

WSXACMLAssertionType class

This interface defines abstract methods that are implementable in the concrete classes that extend it. The concrete implementation provides the functionalities used for the construction of *Requirements* and *Capabilities* child elements.

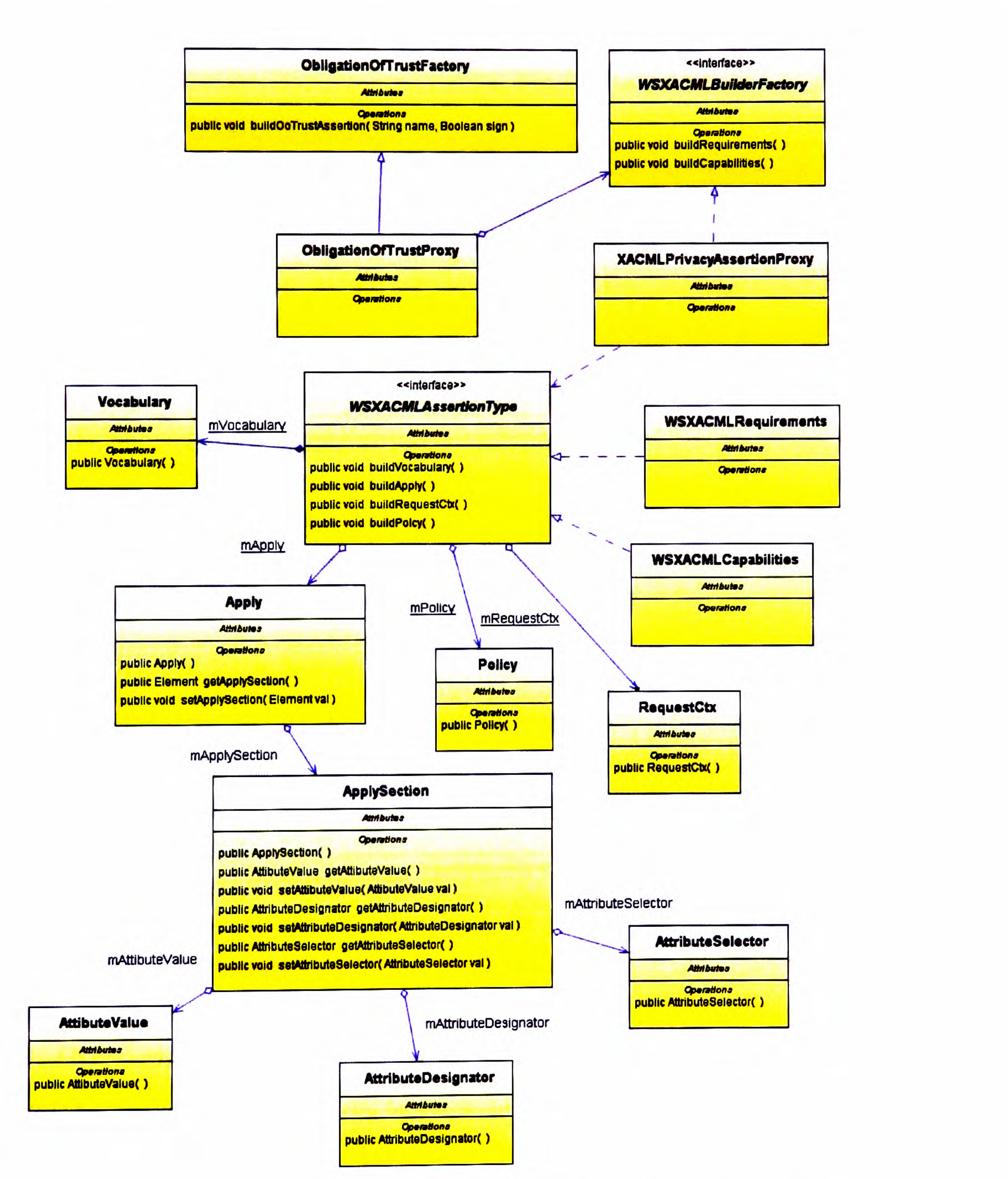


Figure 9.8 WS-XACML Module UML Class Diagram

WSXAMLRequirements class

This is a concrete class that implements WSXACMLAssertionType, which provides convenient logic functionalities for building the elements contained in the *Requirements* section of XACMLPrivacyAssertion.

WSXACMLCapabilities class

This is another concrete class that implements WSXACMLAssertionType, which provides convenient logic functionalities for constructing the elements contained in the *Capabilities* section of XACMLPrivacyAssertion

WSXACMLBuilderFactory class

This interface defines abstract methods that are implementable by the classes that extend it, which provides functionalities that are used in building the OoT message container.

XACMLPrivacyAssertionProxy

This class implements WSXACMLBuilderFactory and is used by ObligationOfTrustProxy object to build the XACMLPrivacyAssertion contained in the ObligationOfTrust container.

ObligationOfTrustFactory class

This abstract class defines the abstract method that is used to finally build the ObligationOfTrust container.

ObligationOfTrustProxy class

This class is the concrete implementation of the ObligationOfTrustFactory, which is used to build the ObligationOfTrust container carried as part of the SOAP Header message context.

9.4 How It Works

The following section briefly describes how the XTMAI works in a typical distributed environment, and demonstrates the interactions between the core participants.

9.4.1 Overview

The XTMAI system has two security handlers, implemented as parts of the XACML PEP and Trust Negotiation Handler (TNH) respectively. The PEP security handler intercepts all access request attempts made to the targeted resources, determines whether access control is required, and takes appropriate actions. The TNH security handler is used for the OoT negotiations, based on the SOAP request-response message exchange pattern. It is assumed here that the web service client (WSC) has performed authentication at its origin by obtaining a token handle, and a web service message is constructed with it. The authentication service of the WSC runs in the same runtime environment as XTMAI that protects the WSC's attribute information, and shares the same public key certificate.

The token handle serves as a *reference link* to the WSC's attribute store (PIP from XACML view point) on the one hand, and part of the subject descriptor in the *XACML Request* context on the other. The token handle which does not explicitly expose the WSC's information authenticates the fact that indeed the WSC comes from the token attesting origin and it is verifiable. In a distributed sense, this phase provides a mutual authentication between the WSC and the WS. Figure 9.9 demonstrates the interactions between the core participants in the XTMAI infrastructure. The following section describes the steps involved in the interactions.

Step 1: The WSC initializes and sends a web services request message to the WS endpoint, together with the token *123456789@xtmai.tcbportal.co.uk* asking for a price list. The PEP handler intercepts this message and performs appropriate security checks.

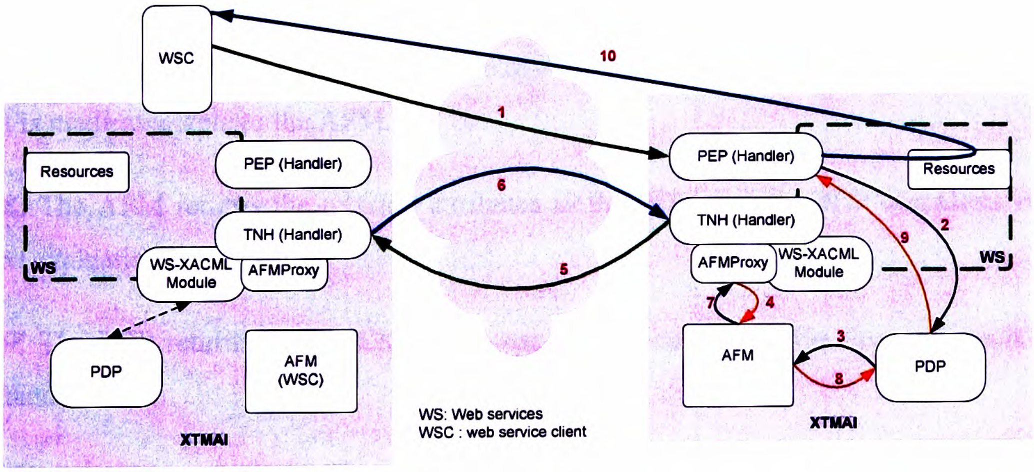


Figure 9.9 XTMAI Core Components' Interactions

Step 2: The PEP handler constructs a *XACML request context* with *123456789@xtmai.tcbportal.co.uk* as the subject attribute, *http://tcbportal.co.uk/pricelist/* as the resource attribute, and **read** as the action attribute, and presents it to the PDP.

Step 3: The PDP in an attempt to evaluate the *request context* against the *local policy* could not be completed due to the missing subject's attribute value(s) which must be provided in order to complete the decision process. The PDP uses the *AttributeDesignator* contained in the *local policy* to identify the required attribute type, and then calls the AFM to supply the missing attribute(s).

Sept 4: The AFM passes the request to the AFM proxy. The AFMProxy uses the WS-XACML module to construct a *XACMLPrivacyAssertion* containing the missing attribute in the *Requirements* section, plus other privacy requirements, and defines what this WS is willing and able to do for the WSC in the *Capabilities* section.

Step5: The TNH in the WS endpoint conveys the XACMLPrivacyAssertion using the OoT protocol in the SOAP header to the WSC's XTMAI endpoint. This is the initial privacy assertion advertisement in the context of OoT protocol message and can only be done in the NoB context.

Step 6: The WSC's TNH passes the received context to the WS-XACML module for processing, and depending on the outcome sends an OoT response context message. *Note that steps 5 and 6 can be repeated as needed until the WSC and WS exchange a SAO context, followed by the actual attributes.*

Step 7: The WS TNH returns a final response to AFMProxy containing the missing attribute(s), which is made available to the AFM.

Step 8: The AFM returns the missing attributes to the PDP, and the PDP completes *request context* evaluation.

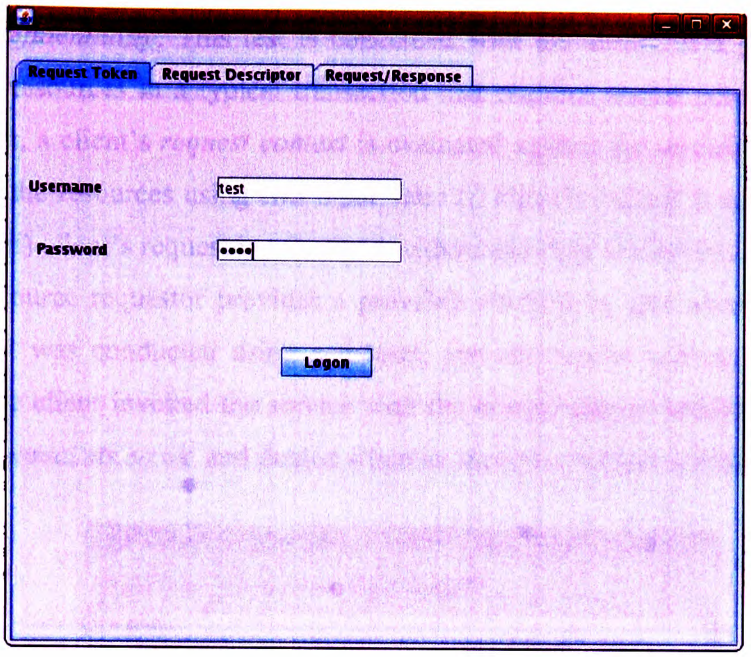
Step 9: The PDP returns the XACML *response context* containing the decision result to the PEP handler.

Step 10: The PEP interprets the decision results, and enforces the decision by either returning the price list to WSC or a denied access response.

9.5 Technical Validation

The validation phase of software is an important aspect that determines whether an application meets its requirement specifications. In other words, it is necessary to verify and validate some measurable quality attributes of a system [172]. In doing the validation test, the limitations imposed by a research environment were recognized, plus the difficulties in getting meaningful results. Nonetheless, it is important to perform some validation tests to substantiate some aspects of the software system thereby validating this research work. The aim of the validation is to verify and assess the software against the objectives and questions addressed by this study. A requirement-based approach [172] was chosen to perform two types of tests, namely, functional and performance tests. Although there is no explicit specification on performance issues, it seemed necessary to consider these because of the intensive computing involved in the processes. Both tests require executing some functions and examining of their inputs as well as corresponding outputs.

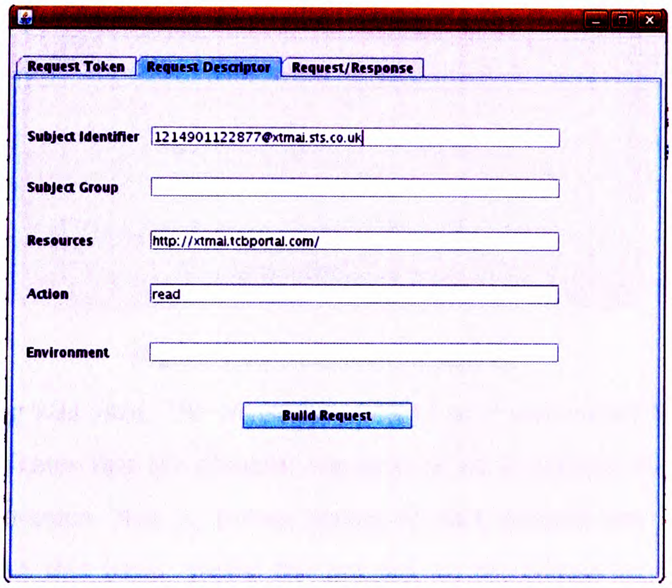
Figures 9.10, 9.11, 9.12 and 9.13 show different views of the User Interface (UI) test bed used in carrying out the various validation tests. On the whole, the activities and behaviours monitored are characterized by inputs initiated by a client, negotiation between a client and a service where appropriate, and output responses. In simple operation, a client initiates a resource request by first performing authentication with its domain STS, using a username/password pair as shown in figure 9.10. The STS authenticates the client and vouches for the client by issuing to it a security token, i.e. 1214901122877@xtmai.sts.co.uk as depicted in figure 9.11, the client uses the issued token to authenticate requests to the service. The service security handler, in this case the PEP, constructs the appropriate request context illustrated in figure 9.12 and submits the request to the PDP. The PDP issues a response context based on the outcome of the access control decision as shown in figure 9.13. In performing these tests, some of the actions are manually triggered in order to facilitate aspects of the interactions. The various validation tests are described in appendix C, and a more detailed discussion of the interpretation of the tests results will be provided in chapter 10.



Username test

Password ••••

Figure 9.11 XTMAI Test bed UI



Subject Identifier 1214901122877@xtmai.sts.co.uk

Subject Group

Resources http://xtmai.tcbportal.com/

Action read

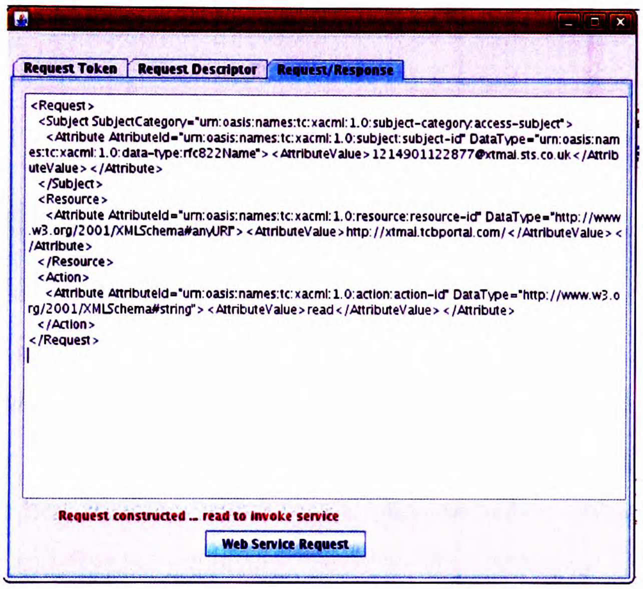
Environment

Figure 9.10 Request Descriptor UI

9.5.1 Functional Test

A set of functional tests was carried out to verify and validate the software against the research objectives and the main research question discussed in chapter 1. The range of tests was carried out using realistic datasets and was done in phases.

Test Case 1 - Confidentiality: This test is concerned with the ability of a party to preserve confidentiality of resources in a typical transaction that requires access control measures. In performing the test, a client's *request context* is evaluated against the service *local policy* that governs access to the resources using two input data: (i) client's request is submitted with the right credentials, (ii) client's request is submitted without the right credentials. The *local policy* requires that a resource requestor provides a provable attribute to gain access to a protected resource. This test was conducted using a dataset, and the results indicate that access was permitted when the client invoked the service with the correct subject attribute credential e.g. *1214901122877@xtmai.sts.co.uk*, and denied when an incorrect subject attribute credential e.g. *1214901315128@pcg.org* was used. The transcript of this test is documented in appendix E.1. The outcome simply indicates that the domain: *xtmai.sts.co.uk* is defined in the service *local policy* as an allowable domain, that is, tokens issued by this domain can be accepted, and access can be granted on this basis, whilst the *pcg.org* is not allowable. In this test-case, confidentiality is considered from the service perspective only, and privacy was not a concern for the client. This is so because the *1214901122877@xtmai.sts.co.uk* does not explicitly expose the client's PII, though it does give a clue to the client's domain. The *1214901122877* is a pseudonymized identifier for this client, and its properties are only known to the client's domain.

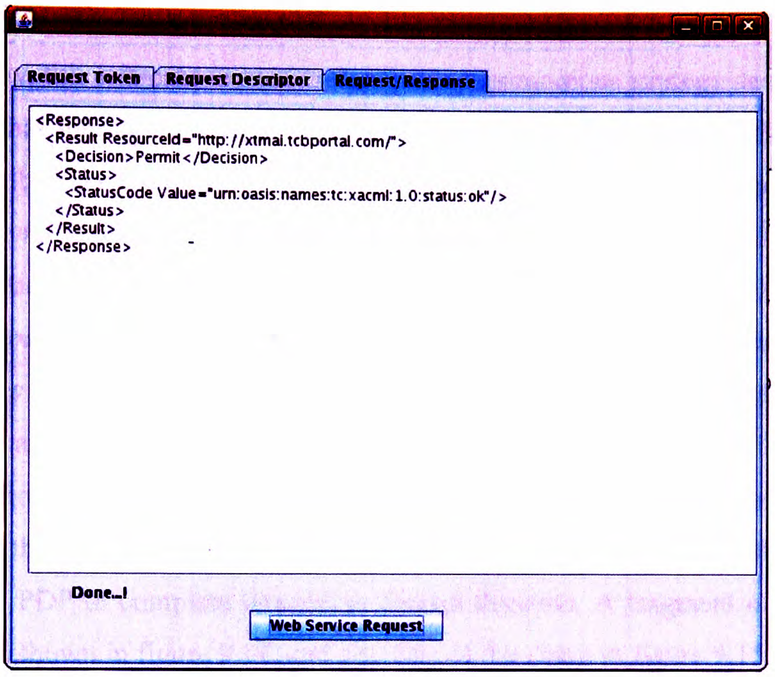


```
<Request>
  <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" DataType="urn:oasis:nam
es:tc:xacml:1.0:data-type:rfc822Name"><AttributeValue>1214901122877@xtmal.sts.co.uk</Attrib
uteValue></Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www
.w3.org/2001/XMLSchema#anyURI"><AttributeValue>http://xtmal.tcbportal.com/</AttributeValue><
/Attribute>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.o
rg/2001/XMLSchema#string"><AttributeValue>read</AttributeValue></Attribute>
  </Action>
</Request>
```

Request constructed ... read to invoke service

Figure 9.12 Request Context UI

Test case 2 - privacy and confidentiality (parsimonious strategy): The test here is concerned with simultaneous protection of privacy and confidentiality between the client and the service.



```
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
```

Done...!

Figure 9.13 Response Context UI

In this test-case, the *local policy* that governs access to the protected resources at the service end requires a requester to submit two different provable attributes before access can be allowed. In this case, the first attribute, which was a token served to authenticate the client in the context of its origin, and filtered requestors based on their domains. The second attribute, which must be submitted by the client before access to the resources is permitted, is considered sensitive by the client and would require privacy protection. Furthermore, the service requires that the resources released to the client be treated with some respect for privacy.

The client and service require a bilateral privacy negotiation in order to determine access to their various resources whilst ensuring privacy. To achieve this goal, the client submits a request to the service with the one out of the two attributes that is considered not sensitive, and allows the service to ask for the second. To validate this case, a complete set of subject attributes were submitted in one-shot; the result indicated that privacy negotiation was not initiated by the service. In contrast, when a request was made with a subject's attribute, privacy negotiation was triggered when the PDP at the service end made an attempt to retrieve the missing subject's attribute(s), and both parties exchanged OoT message contexts; upon successful negotiation, the client returned the missing attribute back to the service before the access decision was completed and a PERMIT response was sent back to the client. The complete transcript of this test-case is documented in appendix E.2.

Additionally, this test-case is used to verify the parsimonious strategy described in chapter 8. In the event that the client's request was not completed by the service PDP due to the missing subject attribute(s), the service sends an OoT context as a NoB describing its privacy assertion (*Requirements* and a set of *Capabilities)* to the client's endpoint. In performing the matching and evaluation operations at the client's side, service's *Requirements* is satisfied by a set of client's *Capabilities* on the one hand, and client's *Requirements* is satisfied by a set of service's *Capabilities* on the other. In response to the service NoB, the client sends back an OoT context containing an SAO message. Upon the validation of the client's SAO, the service sends a corresponding SAO back to the client. The client after validation sends back the OoT context containing the SAML AttributeStatement with the client's attribute, which is returned back to the service PDP to complete the access control decision. A fragment of the service's privacy assertion is shown in figure 9.14, and also that of the client in figure 9.15. The service specified in its *Requirements* that a party must provide an attribute identified by *pcg-group* and issued by *pcg.org,* and information given to a party must not be retained for more than *30* days. In its

```
<Requirements>
  <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
  <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
     <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
       <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
               DataType="http://www.w3.org/2001/XMLSchema#integer"/>
     </Apply>
   <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">30</AttributeValue>
   </Apply>
<Apply FunctionId="urn:oasis:names:tc:xacml.1.0:function:string-one-and-only">
     <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                         AttributeId="pcg-groups"
                         Issuer="admin@pcg.org"/>
  </Apply>
</Requirements>
<Capabilities>
   <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
   <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function.must-be-present">
      <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
         <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
      </Apply>
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">40</AttributeValue>
   </Apply>
</Capabilities>
```

Figure 9.14 Fragment of Service's Privacy Assertion

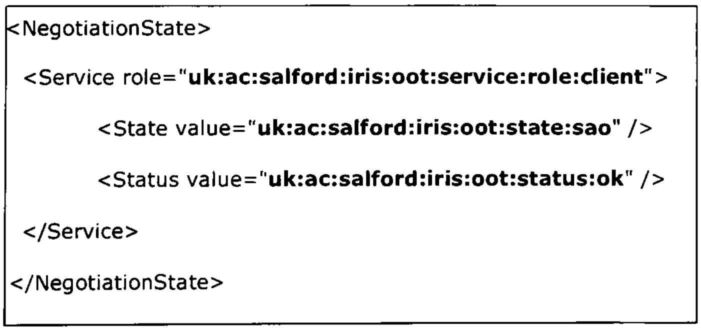
```
<Requirements>
  <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
  <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
      <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
         <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action: max-data-rention-days"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
      </Apply>
   <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">45</AttributeValue>
   </Apply>
</Requirements>
<Capabilities>
   <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
   <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
      <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
         <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action: max-data-rention-days"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
      </Apply>
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">25</AttributeValue>
   </Apply>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
     <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                        AttributeId="pcg-groups"
                        Issuer="admin@pcg.org"/>
  </Apply>
</Capabilities>
```

Figure 9.15 Fragment of Client's Privacy Assertion

Capabilities, it promised to retain information released by a party for not more than *40* days. Similarly, the client's *Requirements* specified that a party must not retain information released by it for more than 45 days; and also in its *Capabilities*, it promised to release its *pcg-group attribute* that is issued by *pcg.org,* and also that information received from a party will not be retained for more than *25* days. The test was extended to substantiate the case where a service's *requirements* are satisfied by a set of client's *capabilities* on the one hand, but the client's *requirements* were not satisfied by a set of the service's *capabilities* on the other. In response to this NoB, the client could not proceed with the negotiation. The client sends a failure notification signifying it is unable to continue with the negotiation. Next, the service PDP returns back an *Indeterminate* response because the access control could not be completed. In practice, the PEP should give a DENY response, because an adversary can exploit the *Indeterminate* response to launch more sophisticated attacks on the service.

Test-case 3 privacy and confidentiality (persuasive strategy): The test-case is similar to test case 3, except that it demonstrates the validity of the persuasive strategy described in chapter 8. The service sends an OoT context containing an NoB message describing its *requirements* and a set of *capabilities*. The service *requirements* are satisfied by a set of client's *capabilities* on the one hand, but the client's *requirements* are not satisfied by a set of service's *capabilities* on the other, i.e. the service's *maximum retention period* exceeds that of the client requirements, i.e. the service specified a 40 days retention period. The client sends back a corresponding NoB message to advertise its *requirements* and *capabilities,* to give the service an opportunity to refine its *capabilities* by offering to advance the negotiation. The service refines a set of its *capabilities* i.e. specifies a 20 days retention period, based on the client's *requirements* and sends back a fresh NoB to the client. The client processes the NoB as described above and returns an SAO response to the service. Upon successful verification of the client's SAO, the service sends a corresponding SAO to the client, and the client returns the attribute values defined in its *Capabilities* element in its policy assertion to the service. The service PDP then completes the access control decision and passes the decision back to its PEP. The PEP interprets the decision and returns its response to the client. The complete resulting transcripts of this test case are documented in appendix E.3.

Negotiation State Management: There are two ways the system tracks the negotiation state, which aids in determining the order and sequence of messages to be exchanged. The OoT container contains an attribute type InResponseTo i.e. InResponseTo="**1214905512917**", which when present indicates that the message is in response to a previous message received from a party. The value corresponds to the value of the ootID attribute of that previous message, i.e. ootID="**1214905512917**" A negotiating party could use the attribute to keep the negotiation state as well as determining the message to send. The alternative approach is to use the <NegotiationState> container contained in the <SOAP:Body> ,which is depicted in figure 9.16,



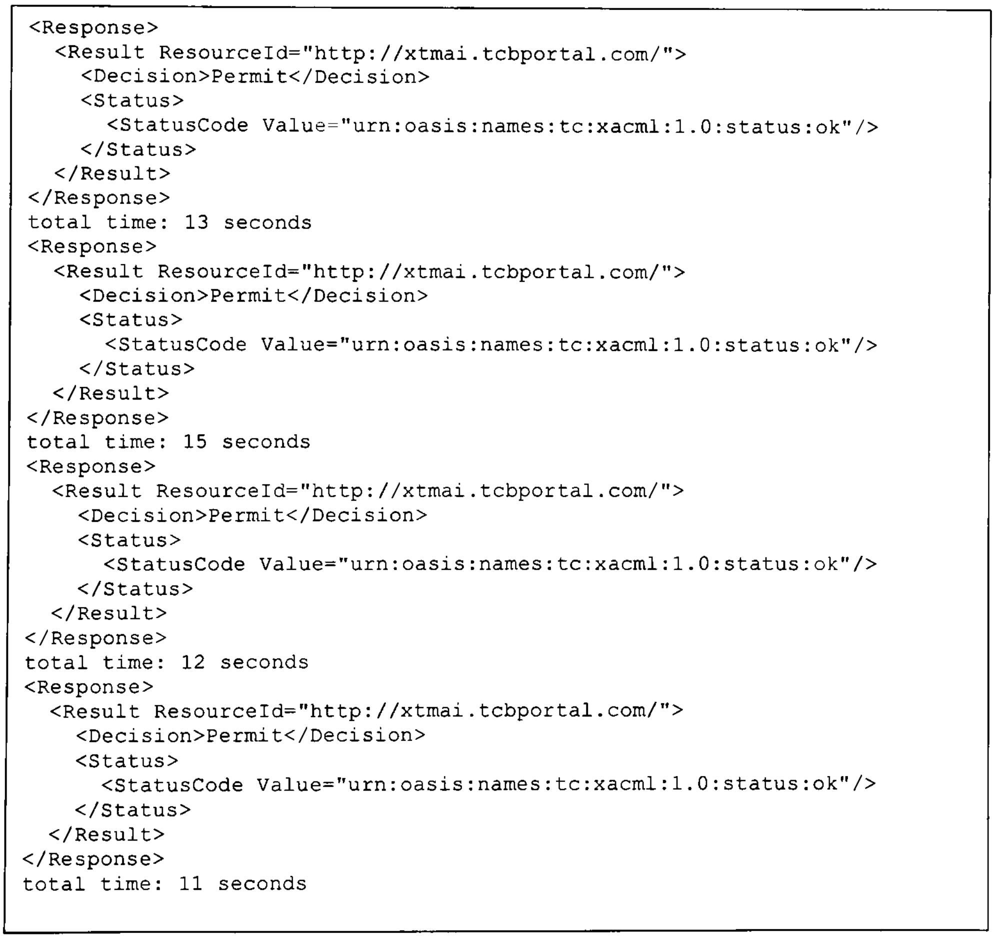
```
<NegotiationState>
 <Service role="uk:ac:salford:iris:oot:service:role:client">
        <State value="uk:ac:salford:iris:oot:state:sao" />
        <Status value="uk:ac:salford:iris:oot:status:ok" />
 </Service>
</NegotiationState>
```

Figure 9.16 NegotiationState Container

and conveys information about the negotiation state. The <State> element describes the type of OoT message context and <Status> element tells about the processing result.

9.5.2 Performance Testing

In this test-case, the round-trip is measured to determine the response time between the various test-case interactions. It is important to mention that the round-trip response measurement cannot accurately reflect the real-world case, simply because there are a number of factors such as network characteristics, the computing processes at both communication endpoints, etc, that must be taken into consideration. These can greatly influence the testing results. Another factor peculiar to the measurement here, is that the client and service are in the same runtime machine. The implication is that some of the factors, such as network latency and other communication characteristics, cannot adequately reflect on the test. Nevertheless, the testing performed can reflect the various matching and evaluations, and the cryptographic operations done at both ends are important factors, which could affect performance. Figure 9.17 shows the response time for four consecutive tests using the same dataset. In this test-case, the client submits an attribute required by the PDP to make an access control decision, and the request



```
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 13 seconds
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 15 seconds
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 12 seconds
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 11 seconds
```

Figure 9.17 Performance Test Case 1

resulted in a PERMIT response. The average response time was 12.75 seconds. The processing in this case involves only the PDP that performed the matching and evaluation at the service endpoint. This is the case because the client submitted the correct attribute information that is required by the PDP to make access control decision.

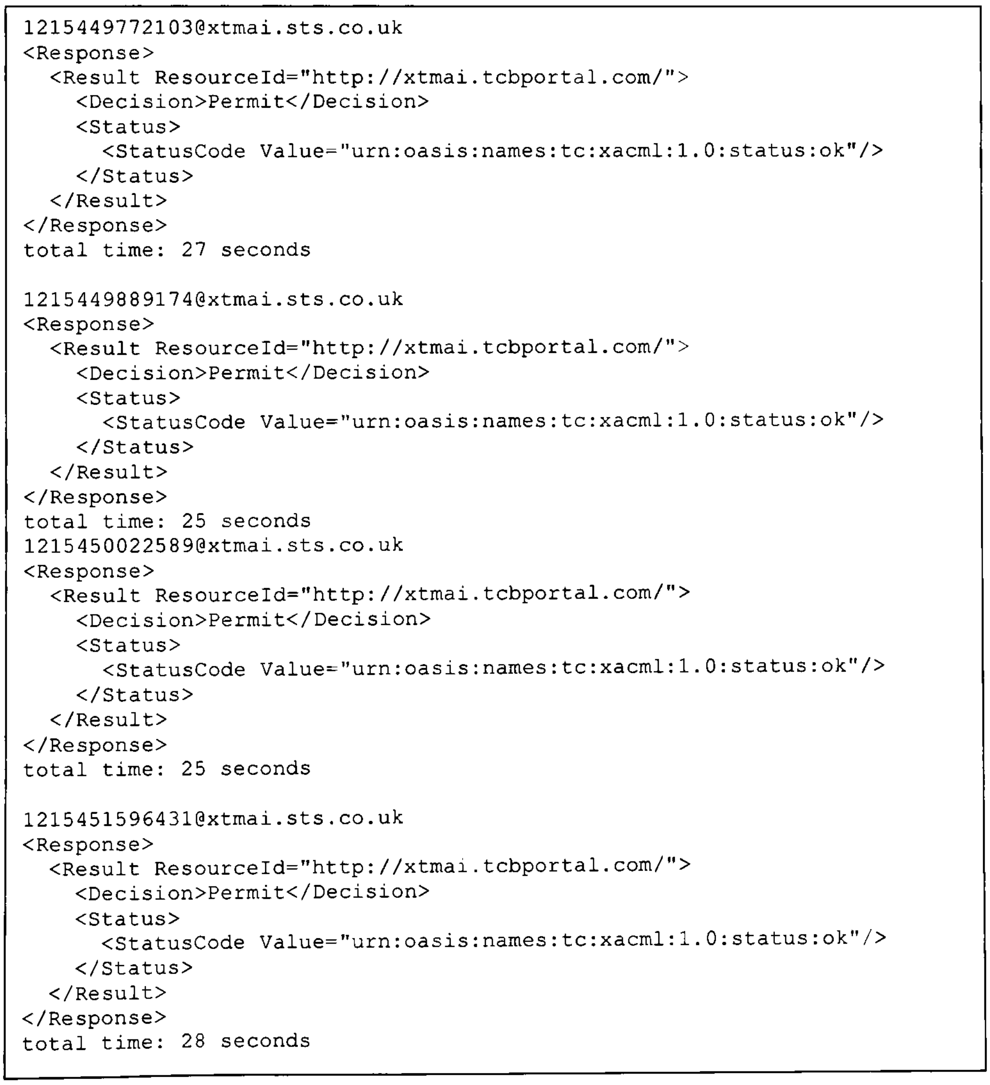
Similarly, figure 9.18 depicts another test-case, where the client submits the two attributes required by the PDP to make an access control decision, and this request resulted in a PERMIT response. The average response time was 21.25 seconds. It is important to differentiate the characteristics of the two tests. Test-case 2 involved more processing because of the need to also evaluate the two attributes of a subject.

```
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 19 seconds
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 23 seconds
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 22 seconds
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 21 seconds
```

Figure 9.18 Performance Test Case 2

Figure 9.19 depicts another performance test whereby the client submitted an attribute out of two attributes required by the PDP to make an access control decision, and privacy negotiation was triggered. The same set of data as in confidentiality test-case II was used, but privacy and confidentiality were processed simultaneously at both ends of the communication, which also involved some considerable interaction between the client and the service. The average response time recorded was 27.25.

As can be seen from the various tests, the performance in terms of response time depends on a number of factors including the availability of computing resources, the number of iterations and the access control input data. Due to the test environment, the resulting response time has not taken network latency into consideration. More discussion of the technical validation will



```
1215449772103@xtmai.sts.co.uk
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 27 seconds

1215449889174@xtmai.sts.co.uk
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 25 seconds
1215450022589@xtmai.sts.co.uk
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 25 seconds

1215451596431@xtmai.sts.co.uk
<Response>
  <Result ResourceId="http://xtmai.tcbportal.com/">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
total time: 28 seconds
```

Figure 9.19 Performance Test-case 3

be given in technical evaluation section of chapter 9.

9.6 Conclusion

This chapter has demonstrated the implementation of a proof of concept, that is, a software system that makes obvious the important capabilities of the artefact developed in this work. The design and development used systematic approaches, and are vital in order to extend the system to real-world application with minimal effort. The details of the architectural modules and base APIs have been discussed. The technical validation has demonstrated that the proposed framework fulfilled its intended purpose, and can help the future implementers of this work in improving the system.

Chapter 10. Evaluation

10.1 Overview

This chapter is an attempt to evaluate the framework developed, and validate it against some measurable metrics. Three constructive assessments are employed in this context to justify the outcome of the work. Firstly, technical validation seeks to test the functionality [51, 173] of the solution against the objectives and research questions outlined in chapter 1. Two techniques are chosen for this purpose, namely experimental and simulation using realistic artificial data. Secondly, performance evaluation [173] measures the average response time during privacy negotiation. Thirdly, critical comparative assessment of XTMAI is carried out against related systems. Trust-X [126], TrustBuilder [120] and Shibboleth [62] are selected for this purpose. The rationale for the choice is that Trust-X, and TrustBuilder shares some similarities with XTMAI in terms of negotiation, and Shibboleth is a widely used distributed access management system within academia [113, 114, 128, 174]. Again, it shares some common characteristics with XTMAI. Six characteristics were selected, namely architecture, protocol and negotiation approach, support for policies and credentials, interoperability, privacy and negotiation capabilities, upon which to base the comparative evaluations.

10.2 Technical Evaluation

In chapter 1, five questions were enumerated that are addressed by this study essentially from the point of view of the question: "how can privacy be preserved when a communicating party's personal information is involved in access control operations?" In context, a communicating party here does not simply refer to a human user, but to any entity, disclosure of whose attribute information may result in embarrassment of any sort. Thus, this study considered end-to-end privacy and confidentiality, using a policy driven infrastructure that makes the solution scalable and extensible. In this section, the author looks back to the main question in an attempt to verify and assess the technical usefulness of the resultant framework.

1 Unauthorized Disclosure of PII: One key step in access control is to ensure the enforcement of the *minimal disclosure principle*. Confidentiality of resources in this context is achieved by the expression of an XACML policy that governs a resource and is matched against an XACML *request context* that describes the characteristics of access

control decision information. A validation test described in test-case 1 section 8.5 of chapter 9 using a set of XACML request context, and the resulting XACML response contexts, validates the capability of the system to filter out access requests based on the *local policy* that governs access to the resources. In this case, access was allowed when the client submitted the right access token, and when the client submitted an incorrect token, access was denied. The resulting outcome validates the capability of the system to restrict access to protected resources based on the XACML *request context* and a *policy* made available to it.

2 Communicating party's active participation: In the scope of the present work, the developed framework provides a novel interface that can allow a communicating party to express its privacy assertions in the *Requirements* section of the policy, which can be evaluated against any request for the retrieval or exposure of the information. Test-cases 2 and 3 described in chapter 9 demonstrate the system's capability in evaluating a communicating party's privacy assertion against another party's capabilities (what this participant is willing and able to do or release) before the party discloses sensitive information. The full test results are documented in appendices E. However, it is important to mention that active participation is not considered from the usability viewpoint, and this aspect is left out as part of further work.

3 Privacy Assurance. One fundamental question this work has attempted to answer is 'how can privacy be guaranteed across security domains of trust?' That is, the provision of a mechanism that will enable a communicating party to evaluate the risks of giving out sensitive information, and determines the degree to which to trust a party in a privacy negotiation. The use of privacy assertions, in which mutually communicating parties express their privacy assertions in *Requirements* and *Capabilities* sections, which can be evaluated against each other, addressed this particular question. The concrete implementation of privacy negotiation strategies further provided a novel framework for gradual and incremental negotiation, in particular a persuasive strategy which allows the parties to refine their *Requirements* and *Capabilities* in a negotiation session. In addition to this, test-case 3 validates the capability from the technical perspective, by generating a SAO in the cause of privacy negotiation between the client and the service. This feature potentially provides a mechanism that can aid in resolving social/judicial aspects of privacy disputes and liabilities amidst other considerations.

4 Remote enforcement of privacy guarantees: The remote enforcement of privacy guarantees is uniquely addressed by the OoT mechanism. It provides a novel interface through its core components: the NoB and SAO, and the protocol, whereby two parties can exchange difficult-to-repudiate digitally signed *obligating constraints* or NoB, which detail their requirements for the release of their sensitive information to a party, and a *proof of acceptance* or SAO, which acknowledges the conditions under which they are accepting another party's information. Additionally, the OoT mechanism provides a framework for the exchange of digitally signed commitments in a tamper-proof manner; technical evidence that can be made available when parties do not conform to their commitments. Test-cases 2 and 3 authenticate these requirements based on the test results documented in appendices E2 and E3. The capability of the parties to receive the SAO before the release of sensitive information uniquely addresses this question.

5 The individual security preferences can be balanced against the legitimate free flow of information naturally using an XACML *PolicySet.* The XACML allows one policy to refer to another policy, or a combination of separately created policies, which can be processed at runtime by specifying an overriding policy and the relevant XACML combining algorithm. With this setup, any conflict resulting from an evaluation of the individual policy rules and that of the super policy (enterprise-wide policy, or country-wide policy) can be resolved into a valid access control decision.

10.3 Performance Evaluation

It is recognized that this work is not concerned with the software engineering aspect; performance issues were not explicitly defined as one of the problems addressed by this work. Nonetheless, given the application environment, it was imperative to undertake a set of performance tests in order to give an indication of the complexity of the processes involved in the framework. As mentioned in chapter 9, the tests carried out have some inherent limitations due to the research environment. Thus related factors in performance testing were not taken into consideration. These related factors among others include optimization of the software algorithms, network latency, computing processing power, etc. Nevertheless, the interpretation of the set of performance tests described in chapter 9 indicates that a number of factors can affect the response time. The resulting average response time shows an increase that is based on the set of input policies, the number of iterations, matching and evaluations, and the cryptographic operations on data; how much these individually affect the systems could only be determined in a real testing environment. Particularly, it is important to know that in each interaction, matching and evaluations must be performed, and perhaps cryptographic operations at both ends. These can greatly influence the response time or throughput. In summary, the performance of the systems will depend on the complexity of the computing operations, network latency, and other network characteristics.

10.4 Comparative Critical Evaluation

In this section, three other systems are compared with XTMAI. The Trust-X and TrustBuilder systems represent novel architectures currently in the trust access control management domain, whilst Shibboleth is a widely deployed federated access management system.

10.4.1 Trust-X and TrustBuilder

Both Trust-X and TrustBuilder were overviewed in chapter 4, and are attribute-based access control architectures. Both are trust access control management infrastructures, in which transaction parties can conduct bilateral and iterative exchanges of policies and certified credentials, to negotiate for the release of resources. These systems are the results of accumulated research outputs in the area of trust negotiation [8, 118, 119, 123, 162], and are well grounded in trust negotiation theories. The following components are used as the basis for the comparison.

Architecture

Both systems are symmetrical with respect to their architecture and are similar to XTMAI though their functional blocks are different. For example, in TrustBuilder, the component that performs security functions is called the *Credential Verification Module*, whilst in XTMAI it is referred to as the Assertion Security Module (ASM), which performs similar tasks. Both modules verify and validate credentials and assertions used in the trust negotiation operations. In the case of XTMAI, it includes methods for creating digital signatures over XML constructs. While the Trust-X and TrustBuilder Compliance Checkers are based on IBM Trust Establishment software, XTMAI uses an open source SunXACML PDP engine [168] for access control decisions. This entails XTMAI using an open standard that has some inherent advantages over TrustBuilder and Trust-X from the viewpoint of the implementation. Given that XACML has enjoyed wide spread support, potentially XTMAI can easily be implemented and/or extended in real world applications with minimal effort.

Protocol and Negotiation

In terms of protocol, Trust-X and TrustBuilder are based purely on a challenge-response protocol [102], as opposed to the request-response message pattern and mutual assertion advertisement used in XTMAI. In their operations, each phase of iteration is either a policy or credential disclosure, meaning that there may be several rounds of iterations (where there are many counter policy exchanges), which have to be executed before reaching a typical negotiation goal. The rationale is that since a party in the negotiation will not be fully aware of what it stands to benefit from the release of its attribute information, it can issue a counter policy in return, which must be satisfied before it can return the credentials that will satisfy the other party. In contrast, in the work described here, the *Requirements* and *Capabilities* of one party are exchanged in one phase of iteration. In doing this, the providing party indicates what it requires from the recipient in the *Requirements* section, and what it is willing and able to do for the recipient in the *Capabilities* section. The advantage is that a recipient is well informed, and so can make a quick decision by weighing the risk of giving its information against the benefit, and potentially would not necessarily issue a counter assertion where there is minimal satisfaction. Additionally, the advertisement of requirements and capabilities offers the benefit of allowing a party to offer alternatives to what is advertised by another party. Once a party is satisfied with the *Capabilities* of a party, and that its own *Capabilities* can satisfy party's *Requirements*, it signs its assertion and conveys it to the participant.

In the case of negotiating strategies, both Trust-X and TrustBuilder have an extensive family of strategies [118] that can be adopted by negotiators in order to find the best way to reach their goals. In contrast, XTMAI implemented two strategies described in chapter 8, but can be extended to accommodate other negotiation strategies.

Policy and Credential Standard Support

In the case of policy expression language, both systems use a proprietary XML-based language to encode their disclosure policies, as opposed to XTMAI that makes use of XACML and its variant WS-XACML (both industry standards) for the representation of disclosure policies.

Similarly, for credential format, both currently support X.509 certificates, unlike XTMAI that naturally inherits all the credentials supported by XACML and SAML including X.509 certificates, SAML assertions, etc.

Interoperability

Interoperability is important for models targeted to open environments. The underlying concepts that drive the systems are yet to be formalized and standardized. The implication is that it is potentially difficult to adapt them in real-world scenarios, except for their proprietary uses. In contrast, XTMAI is based on the XACML and SAML models, which makes it extensible, and more practicable and easily adaptable.

Obligation Capabilities

Given that privacy is a socio-economic problem having contractual properties, it requires a technical solution that derives its concepts and message exchanges from open standards, and a business solution that draws its trust from the enforcement that is provided by the legal and regulatory infrastructure. Consequently, privacy protection entails enforcing privacy obligations that reflect the basic privacy principles [48]. This shaped the development of a mechanism that allows privacy negotiating participants to express such contractual obligating constraints and generate too difficult-to-repudiate technical evidence. It is an attempt to fulfill most of the basic privacy obligation principles. Particularly, this framework merges a technical solution with possible social/judicial aspects for better security assurance in distributed open system transactions. TrustBuilder and Trust-X never considered this important aspect of privacy.

Privacy Capabilities

In Trust-X and TrustBuilder, privacy is tackled from a different perspective than described in this thesis. Arguably, it can be concluded that Trust-X and TrustBuilder consider privacy from a confidentiality point of view, which does not reflect important basic privacy characteristics. In this framework, there is a greater emphasis on foundational privacy, which is addressed in a range of ways, including through the use of pseudonymity described in chapter 7, *XACMLPrivacyAssertion,* and the OoT protocol described in chapter 8. In addition, privacy is considered based on the recommendations of FIPs, and the inclusion of the P3P standard into the solution, potentially making it simpler for enterprises already using P3P to plug in.

Although the Browser Based Trust Negotiation system [127] addresses some of the limitations of TrustBuilder, the framework differs in many respects from XTMAI. The BBTN's protocol is similar to TrustBuilder, which involves disclosure of a policy in an iteration and credential in a separate iteration. In terms of privacy protection, enforcement of privacy and obligating constraints were not considered by the author. The concepts of NoB and SAO described in chapter 8 help in the enforcement of obligating constraints, and provide a mechanism for strong binding of the two statements of commitments, i.e. *Requirements* and *Capabilities,* that define transaction parties' actions, expectations, offers and acceptance of one another.

10.4.2 Shibboleth

Shibboleth is one of the first federated access management to be implemented and has had significant impact in sharing users' information in a secure and trusted manner. The purpose of the information sharing is to facilitate authentication, authorization, content personalization, particularly in enabling single sign-on across autonomous boundaries of trust. Shibboleth is based on open standards, essentially SAML and has no explicit support for XACML. It has two asymmetric distributed subsystems namely: Identity Provider (IdP) and Service Provider (SP). In a normal process, users' information is sent from a home origin IdP to a target SP. In the comparison between Shibboleth and XTMAI, the same components are employed as follows.

Architecture

The Shibboleth infrastructure comprises two main distributed software systems that are asymmetric in characteristics. The IdP serves as a SAML Authority that issues assertions to a user's browser that has successfully performed authentication with it. Based on the assertions, the user can connect to a resource site of which the SP software is the gatekeeper. The SP software can decide to grant or deny access to the user, or optionally ask the IdP for more attributes of the user. The SP software uses an in-built callback mechanism to ask the IdP for the attributes, and in turn, the IdP uses an in-built ARP to decide whether to release the attributes to the requesting SP. This is unlike XTMAI, which is symmetrical when deployed at both endpoints and which allows both the user and SP to conduct iterative and bilateral exchange of resources. Nevertheless, it can be argued that the callback mechanism is designed to achieve the ***minimal disclosure principle*** and further protects the user's privacy in the Shibboleth case.

However, the dissimilarity with this framework is the lack of mutual interaction and the ability to negotiate for the release of resources. In Shibboleth, the SP sends an *AttributeRequst* message, which does not contain the *purpose* for which the SP is requesting more attributes. In contrast, XTMAI uses an *ObligationOfTrust* context which describes privacy requirements (e.g. the purpose, retention period, etc.), which the SP is willing and is able to do for the client, as well as what it expects from the client. In XTMAI, the client responds with another *ObligationOfTrust* context, which may be an NoB or SAO depending on the message, and the interaction can continue until they can mutually agree to release their sensitive resources.

Protocol and Negotiation

The Shibboleth uses a synchronous request-response message exchange pattern (MEP) which is essentially based on HTTP browser protocol. The XTMAI uses the same synchronous request-response protocol, particularly the SOAP MEP, but defines a service interface to take into account other application level protocols, including HTTP browser. The Shibboleth infrastructure has no mechanism for message level negotiation as presented in this thesis, and so is a unilateral access control paradigm.

Policy and Credential Standard Support

The Shibboleth uses *ARP* and SAML assertions in the full life-cycle of access control of service resources and users' attributes. In chapter 4, the defect of the *ARP* structure was discussed, and its lack of support for foundational privacy principles. Nevertheless, in terms of credentials, it supports many of the credential profiles defined in the SAML specifications. XTMAI uses XACML, WS-XACML and SAML assertions for the full life-cycle access control of service resources and users attributes. The WS-XACML extends the capability of native XACML to enable robust, scalable and mutual privacy assertion negotiations. The WS-XACML has a natural way to allow the SP to convey its privacy statement to a client, using the P3P policy framework in a web services interaction. Given the above comparison, XTMAI is more flexible in terms of access control policy expression than the Shibboleth, and potentially covers a wide range of access control use-cases.

Interoperability

Both systems have the characteristics of interoperability since they are based on industry standards. In the case of Shibboleth, it is a distributed system with well defined semantics and syntax, and has been implemented in many software languages and platforms. In contrast, XTMAI still has to undergo some further analysis in system design and protocol refinement to make it more mature.

Obligation Capabilities

It has been established by this thesis that privacy is a socio-economic problem with contractual properties. These properties are backed up by law, legislation and principles that must be considered by privacy aware systems [48]. In this framework, privacy protection has been considered from this background, which led to the development of a mechanism that allows the expression of contractual obligating constraints, to deal with privacy obligations between transaction participants. Additionally, the OoT makes it possible to generate too difficult-to-repudiate technical evidence, in an attempt to address remote enforcement of privacy. This potentially merges a technical solution with possible social/judicial aspects, for better privacy assurance in distributed transactions. Shibboleth never viewed privacy from this perspective at all.

Privacy Capabilities

Both systems have privacy as a primary goal, except that they achieved it in different ways. In the Shibboleth scenario, privacy is achieved by means of pseudonymity and the *ARP*. In XTMAI, privacy is achieved by pseudonymity, WS-XACML assertion and the OoT mechanism. Arguably, Shibboleth's privacy mechanism did not take into account the basic privacy principles, and it could be considered as satisfying confidentiality requirements rather than privacy requirements, as has been distinguished in this thesis. Again, the privacy solution is considered from a wider perspective including privacy law, legislation and principles as presented in chapter 3; this Shibboleth is yet to do. The use of WS-XACML and the OoT protocol is a significant improvement in the treatment of privacy, compared to Shibboleth's privacy provision.

To sum up the comparative assessment, it is important to highlight that the remote enforcement of privacy and confidentiality in distributed environments is critical and complicated. The development of OoT with its components NoB and SAO has attempted to address the remote enforcement of privacy obligating constraints, and is a key step towards the achievement of privacy assurance. Transaction parties can assert their requirements and willingly accept the assertions of others about their requirements and reach mutual agreements on their capabilities. This is a unique contribution to the body of knowledge in the general area of security and privacy.

10.5 Conclusion

This chapter has evaluated the XTMAI framework against three other approaches that address parts of the problem of maintaining privacy in access control. It has shown that the framework has the potential to improve privacy in distributed environments. This work has therefore met the objectives defined in chapter 1, and addresses a need that was identified through the secondary research reported in chapters 4, 7 and 11.

Chapter 11.Conclusion and Future Work

11.1 Overview

This thesis has described work carried out to investigate how privacy can be protected when a communicating party's identifiable attribute information is involved in access control operations. The work undertaken has successfully followed established methodological approaches in the analysis, design, implementation and evaluation of Privacy Trust Access Control Infrastructure using XACML, and its proof of concept, the XTMAI. This work is based on existing and relevant Internet standards, specifications and security enhancement technologies. The goal is to ensure that the resultant technical solution is adaptable, interoperable and extensible. This is the main rationale for choosing the XACML and SAML models, which are industry standards developed to address a number of access management security problems in open environments.

This study commenced with an extensive investigation, analysis and assessment of existing access control management systems [7, 18, 116, 126, 128]. It was felt that policy based security services will address privacy more appropriately than most anonymous systems [85, 88, 175]. An Access management system is a major security service that addresses application level security, by restricting the use of computer resources to authorized users, i.e. entities with provable credentials. It was realized that the rise in cross-organizational transactions requires this important security service which may involve both authentication and authorization techniques.

However, for access control decisions to take place, the underlying system makes decisions based on its local access policy against certain attribute information presented to it. This, in many cases, is in the form of identifiable attributes of the access control actors. It has been sufficiently established by this thesis that one critical item of information that is usually contained in the attribute information, is what is often referred to as PII. This significantly highlights the concern for privacy and confidentiality, which are the primary focus of this work. More importantly, it has been established through the analysis of the relevant literature that privacy options can no longer be completely isolated from the notions of confidentiality and trust, in the sense that in many occasions, they need to work together to achieve the desired privacy security goals.

After a critical analysis and assessment of current systems, it was concluded that existing access control systems often treat privacy, confidentiality and trust independently, on the basis that privacy is the client's concern, while confidentiality is that of the service provider's. This treatment has resulted in access control systems that prevent unauthorized access to enterprise resources, but without adequate consideration for privacy. Overall this thesis has exposed the gap in most existing access management systems which are asymmetrical in architecture, except trust access management systems that have a symmetrical architecture. Nevertheless, trust based access management systems have a significant weakness with respect to privacy and confidentiality highlighted in chapter 4.

The initial findings of the research were fed into the conceptual design of XTMAI presented in chapter 7, which was developed as a result of the detailed analysis of related privacy literature in chapter 3, access management in chapter 4 and privacy enhancement technologies and standards in chapters 5 and 6. In general, the resultant artefact is designed based on the identified weakness in existing systems, and certain factual principles that were evident from an early stage of the work. The following outlined factors were instrumental to the overall development of this work:

1 The unilateral or non-mutual treatment of privacy and confidentiality in most existing attribute-based access control system architecture i.e. client-server. In many of these systems, the client is required to disclose personal information unconditionally to the server, irrespective of whether the service can adhere to the security preferences of the client; otherwise access to the service cannot be allowed.

2 The lack of privacy assurance mechanisms; the client is unable to enforce how information passed to another party can be restricted and used only by the authorized parties and for the intended purposes only. There is an absence of techniques to make an attribute consuming party to convey an acceptance or rejection of obligating constraints imposed by the providing party before the release of attribute information.

3 Current systems overlook server side privacy. Where existing systems consider privacy [11, 18, 123], they do not draw on the fundamental principles underpinning privacy protection [14]. However, service providers on the other hand may have certain provable attributes that the client would like to see, which disclosed to arbitrary strangers might leak important business information.

In the development and design of the technical solution, a security threat modelling in chapter 7 was done to supplement and validate earlier assumptions made about privacy, confidentiality and trust. The essence was to uncover privacy vulnerabilities from an infrastructure viewpoint, determine the likelihood of threats, and establish the applicability of privacy protection tools within the context of access control operations in distributed environments. The analysis of how distributed XACML model actors can participate to enable bilateral request-response message exchange pattern was undertaken. The investigation and analysis of suitable options for the remote enforcement of privacy led to the conceptualization of the OoT protocol. It is considered a significant development that provides a concrete approach to enhancing privacy assurance with strong binding of obligating constraints, and its development led to refereed publications [31, 47].

11.2 Strengths

This framework is based on broadly accepted industry security standards i.e. XACML, SAML, etc, and it combines a variety of other concepts, i.e. trust negotiation, fair information principles, etc, to address important privacy and confidentiality concerns in a mutually bilateral manner. It considers open systems characteristics, separating the authentication and authorization phases of access control in a fashion that enables privacy negotiation between two communicating parties. Given that most existing systems overlook mutual treatment of privacy and confidentiality, the architecture being symmetrical, enables the service provider and client to evaluate the risks of transactions that involve identifiable or contractual information that is of privacy value, before the exchange of such information. Being based on industry standards that have the spirit of extensibility, the work is extensible to the point that potentially it can be used to address other foreseeable privacy and security threats.

An additional strength of this work over traditional TN is that it has the potential to reduce the number of iterations between negotiating parties. This may mean a significant improvement in the average response time since both *Requirements* and *Capabilities* can be conveyed in a single payload rather than separately. The overall design of the XTMAI makes it stand out from among other access control infrastructures. The emphasis of XTMAI in the treatment of privacy and confidentiality in which XACML, SAML and P3P are combined makes it adaptable even to existing systems. Web service providers can plug their existing P3P policy statements written in XML into XTMAI, and use them for privacy negotiation with minimal effort.

Lastly, a mechanism that demonstrates noteworthy improvement in the provision of privacy where "difficult-to-repudiate" services are vital is essential, to assure each communicating party in distributed transactions that their information will be used in accordance with their wishes.

11.3 Benefits

The central benefit of this work is that parties in open systems transactions can evaluate the risks involved in sharing their sensitive resources, whether they are computer resources, policies that govern access to resources, contractual agreements or identifiable attribute information. The risk evaluation criteria using trust negotiation depend less on public key infrastructure, and more on provable attributes or abilities of the participants. The negotiations, cautiously based on gradual and incremental revelation of the provable attributes that enable the building of trust, help the parties involved to compute the trust threshold and make informed decisions about the release of their resources. Given that the framework provides the mechanics to negotiate 'difficult-to-repudiate' technical evidence, it has the potential to increase the level of trust and confidence between the communicating parties and may reduce the liabilities of regulated organizations.

The use of two related policy frameworks allows separation of duty in the treatment of privacy and confidentiality. This separation is critical since some enterprise resources, though sensitive, may not have privacy values, since privacy is guarded by laws, legislation and principles that are obviously influenced by other factors including culture, religion, as well as regional peculiarities.

11.4 Vulnerabilities

Both the XACML and SAML specifications named known security threats inherent in open networks, and described a number of countermeasures to address them. These threats include amongst others, man-in-middle attack, message modification, message deletion, message insertion, session hijack, replay, impersonation without re-authentication, probing, inferential attacks, etc, and are innate open and distributed security threats. It is important to mention that XTMAI by its nature inherits these vulnerabilities. Whilst most of the threats have existing countermeasures, i.e. session management, XML encryption, XML signature and TLS, threats such as probing and inferential attacks are specific to trust based access control models. Descriptions of some of these are listed below with suggestions for possible countermeasures.

Unauthorized disclosure of access control decision information: In many access control operations, the undesirable disclosure of the properties of the access control actors may constitute major security breaches. In other words, the correctness or reliability of the properties of the various actors is as sensitive as the protected resources. For example, the disclosure of an access control policy may expose important business information that can be exploited by malicious parties. Threats in this regard include messages in transit, undesirable disclosure due to elevation of privileges. Generally, there are existing techniques for addressing this kind of threat and they are encouraged here. Nevertheless, XACML policy language has extensive ways to enable minimal policy disclosure, which can limit the undesirable disclosure of information about the attributes of actors. In fact the OoT, when used correctly, can mitigate these types of vulnerabilities by including minimal *requirements* and *capabilities* that are sufficient to advance the trust negotiation.

Message Replay: This aspect results from a malicious party's ability to record and replay legitimate messages between transaction parties, which can also lead to denial-of-service attacks. It is possible for a malicious entity to intercept OoT messages and use them to launch an attack. This kind of attack might be addressed by ensuring that messages are fresh, which can be made possible by using the *issuerInstant* attribute value of the SAML assertion and setting an allowable time interval i.e. validity of assertions. This is a notable use of short-lived security tokens.

Message Insertion: This threat is common as an adversary can inject privacy assertions in the sequences of OoT between negotiating parties. Nonetheless, there are existing solutions that can mitigate this attack, including mutual authentication of every message to ensure its integrity.

Message Deletion: An adversary can delete OoT messages in the sequence between two OoT actors. Existing message sequence integrity safeguards can be utilized to mitigate the threat.

Message Modification: The integrity of OoT messages is critical, if a malicious party is able to modify their content, it will alter the negotiation outcome, which will invalidate the negotiation. To mitigate against this, message integrity techniques should be utilized.

Negotiation Outcomes: The handling of negotiation failures should not reveal information that is sufficient for an adversary to launch an attack. In the case of a native XACML *authorization decision* of "NotApplicable" the recommendations described in [37] apply.

Cyclic Effect: The persuasive negotiation strategy has the tendency of introducing an unnecessary cyclic effect between negotiating parties, which can be exploited for denial-of-service attacks. This effect can be resolved by setting a limit to the number of NoB contexts that can be exchanged using the inbuilt InResponseTo attribute to monitor the interaction state.

Probing and Inferential: A malicious party can mount bogus negotiations in order to infer or extract information from another party. In this case, the threat can potentially be mitigated by crafting the XACML policy in such a way that a sensitive negotiation policy can be protected against undesirable disclosure [30]. This can be done by using the extensive filtering mechanism to filter inbound requests from the top-level of the policy. Using the *Target* scope in the *Policy* element or the *PolicySet* element a knowledgeable policy writer can express first level filters that filter out untrusted domains.

It is important that appropriate countermeasures be taken to ensure the security of the infrastructure and its components. Resource policies are themselves as sensitive as the resources they protect. Unnecessary disclosure of an access control policy can reveal the details of subject descriptors, which a malicious party can use to establish how to get unauthorized access.

11.5 Limitations

Nevertheless, the XTMAI system has a number of limitations. A few of these limitations are inherent due to the need for a PKI, others are imposed by the security standards upon which it was based, as well as the lack of formalization of the OoT protocol itself. A few of these limitations are presented in this section.

A private key stored in a party's computer, protected by a username/password has a few flaws:

- The party can only sign data on that particular computer; for a server, or an application, this is not an issue. For a human user it is obviously an issue.
- The underlying security of the key depends entirely on how secured the computer is. It is possible to copy the private key.

A malicious party who gains control over a party's computer may substitute a user application with a malicious one. This will potentially, tricks a party into signing any data. This limitation could be solved by ensuring that applications calling the signing software are properly authenticated.

However, a more secure alternative is store the private key in a tamper-resistant device, such as smart card. But, smart card technology introduces an additional overhead in terms of the need to manage its life-cycle, including activating it with personal identification number (PIN). Although with the smart card, it is potentially difficult to copy the private key, the card can be lost easily.

The XTMAI requires token issuing authorities to be PKI enabled, so that the tokens can be verified and validated using public key cryptography. Public key cryptography is known to be a time and computing intensive process; consequently, it can take time to perform cryptographic operations including the validation of PKI certificate path.

Privacy and confidentiality as presented in this thesis assume that a form of primary trust model exists. There is a need to establish how one participant can come to believe that a given key is exclusively associated with a unique identity, so that the key can be trusted to sign or verify the assertions of other integrity and non-repudiation services. The means to establish relationships between the transaction parties may require a combination of two or more trust models.

Digital signature schemes do not provide certainty about the time and date, at which the signature was applied by a party. The implication is that a signer may use a timestamp (i.e. a clock that is not synchronized with a time server); this may give a malicious signer the lean way to manipulate the date and time the signature was applied. This limitation can be mitigated by using a trusted time service.

The XTMAI infrastructure relies on two policy types: a native XACML policy and a WS-

XACML policy assertion to simultaneously deal with privacy and confidentiality. The two policy sets must be linked somehow in order to enable privacy negotiation and the exchange of policy assertions. This approach may have undesirable consequences for the users. However, to overcome these limitations, it is possible to have one static native XACML policy set, and then construct a WS-XACML assertion dynamically. The implication is that other policy mappings such as the P3P-XACML mapping must be done in a static mode.

One key benefit of OoT is the capability to generate complete hard-to-repudiate technical evidence of distributed transactions. Nonetheless, it was recognized that there may be some limitations from the legal point of view. That is, in some situations, a "non-repudiable signature" is not sufficient evidence for legal cases, although it can be considered important. Other factors include 'how active' was a participating human user, in deciding to sign the other party's privacy obligating constraints or whether software automatically generated a signature on behalf of the human user; did the human user understand the complexity of privacy obligating constraints, and under what type of user interface (UI) did the user click "Agree" resulting in the signature, are critical limiting factors. Obviously, there are significant legal issues involved in providing digital evidence for privacy solutions, the attempt to provide tenable evidence should not be considered a substitute for proper consideration of legal issues; the debate about what is legally admissible is outside the scope of this work. However, the EU in [176], clarified the legal status of digital signature, and guidelines are defined in order to ensure their basic legal recognition and validity.

The other limitation of this work is the lack of formalization of other components of OoT such as the negotiation strategy. Though, the OoT protocol is an extension of the SAML request/response protocol, there are quite a number of limitations imposed by the OoT negotiation by itself. The exchange of messages, order and sequence of the messages may be limited by the negotiation strategy the parties adopt. The fact that the framework incorporates a few of the negotiation strategies may limit its use in some complex privacy negotiations. Similarly, the absence of a formal usability study is a drawback,

11.6 Summary of Major Contributions to New Knowledge

The work described by this thesis has made contributions in a variety of domains. In the earlier survey of the literature, it was supported that a asymmetric approach in access control systems in distributed environments cannot guarantee the privacy of the client's attribute information, and with other collaborators a paper was published on how XACML can fit into trust negotiation [30]. This paper is based on a critical assessment of trust negotiation systems that are currently neither standardized nor based on any known standards. As the work progressed, the analysis of alternative architectures to existing ones resulted in a further publication [45], in which a potential infrastructure was proposed for addressing the privacy problem. Furthermore, in the course of the work, a use-case based on an e-procurement platform for the construction industry was developed. The modelling, deconstruction of the access control participants in distributed sense, and the critical analysis of the various interactions and trust contexts, which formed part of chapter 6 resulted in yet another publication [46].

An infrastructure for the bilateral privacy negotiation that is symmetric in architecture has been developed, and a proof of concept implemented that validates the concepts developed over the duration of this work. Looking back at the literature reviews in chapters 3 and 4, it is evident that very few systems were in place or had viewed the solution from the same perspective that underpinned this work. Therefore, it can be concluded that the work described by this thesis in many ways supplements work carried out by others, plus providing an insight on the current status of existing systems. It has provided an alternative approach in addressing the privacy problem with a viable technical framework that attempted to provide a solution to it.

Significantly, in the domain of privacy and confidentiality, this work has made a concrete contribution in the field of Internet security with the development of the OoT, which is potentially a valuable step in protecting privacy across autonomous security domains. The author developed the concepts surrounding the design of its protocol components, whilst the contribution of other collaborators in the full realization of the OoT is acknowledged; all the foundational work was the author's. A more complete description of the OoT concepts was disseminated in recent publications [31, 47], which details the characteristics of the problem it attempts to address and the benefits of the approach. Finally, another noteworthy contribution has been in the area of security architecture design and development. The adoption of several Internet Security standards and putting them into a logical architecture is in itself a pragmatic contribution to the body of knowledge.

On the whole it is considered that the work described by this thesis has established new knowledge, made valid and unique contributions theoretically and practically to the field of privacy and security. In particular, the results of this research have already been made available to the academic community in publications, conferences and workshops.

11.7 Suggestions for Future Work

Besides addressing the above mentioned limitations and deficiencies of this work, there are more grey areas in which to extend the framework. The OoT protocol would require enhancement to support other access control requirements. For example, in some cases, policies governing the authorization of high-value security transactions may require simultaneous actions by more than one subject or actor. In the case of distributed environments involving more autonomous security domains, if the subjects that are required to execute the action are from different domains, and the attribute information required to perform the actions are considered with respect to privacy, it may require simultaneous but separate negotiations to obtain the various attribute information in order to perform the joint access control actions. The above case requires enhancement of the OoT protocol by way of devising strategies to handle such complex cases.

Furthermore, the introduction of ontology and semantic web technologies into the OoT mechanism could provide the basis for advanced intelligent reasoning, and the composition of privacy assertions that are purely constructed dynamically, based on certain risk threshold calculations. The idea is to provide transactional parties with more sophisticated ways for dynamic and intuitive handling of their *Capabilities or Requirements* during privacy negotiations.

Additionally, it is recognized that usability is an important aspect of an Information system as such, a formal usability study would be necessary to evaluate the work from a user's perspective.

11.8 Conclusions

This work has provided a unified framework for protecting privacy on the one hand and confidentiality of service outputs plus service meta information on the other. In the design and development of the work, privacy and security characteristics were examined in the context of access control systems, relevant legal and regulatory literature were reviewed as well as critical appraisal of current systems. It is been made clear that this work derived its concepts and message exchanges from broadly accepted standards. The proposed technical solution is driven by the need to support privacy protection on the basis of the legal and regulatory frameworks. To the best of the author's knowledge, none of the current systems provide a mechanism for the remote enforcement of privacy obligations as presented in this thesis. The implication is that the absence of such an important security service might reduce communicating parties' trust, since it is uncertain that their information will be treated in a manner that respects their privacy.

To the best of author's knowledge, this work is among the first to introduce the remote enforcement of privacy obligations, particularly, the development of the OoT protocol to support its implementation. The OoT mechanism provides an extensible means to support potential social/judicial solutions for security assurance in distributed open systems. The NoB and SAO mechanisms equally provide a standard way for the dynamic exchange of other obligating or contractual documents such as SLAs, BLAs, which provide possible automation of contractual document exchanges, which can improve traditional methods that are time consuming and costly. The OoT mechanism provides significant improvement over traditional trust negotiation; it has the potential to reduce the number of iterations between negotiating parties compared to other mechanisms that were examined, so that the effect of negotiation response-time is reduced significantly since both requirements and capabilities can be transmitted in a single payload rather than separately.

Nevertheless, the framework described by this thesis has a small number of limitations, which require further investigation and analysis. Some useful suggestions on the areas of improvement have been suggested, which potentially should address some of the limitations mentioned. Apart from these limitations, it can be concluded that this work has met the objectives defined in chapter 1, and addresses a need that was identified through the secondary research and the evaluation reported in chapter 10.

Finally, another innovation of this work is the benefits of the OoT mechanism. More importantly, the OoT provides a mechanism that allows communicating parties to indicate their willingness to accept constraints imposed on their use of information provided by a party. More so, it provides the capability to express what a party is willing and able to do for the other party before they can reveal or exchange sensitive information, which is a momentous improvement in guaranteeing privacy. In other words, the parameters to gauge the level of assurance and/or confidence that should be associated with a given transaction, in reality should depend on the level of trust established by communicating parties themselves. Overall, the development of the OoT concepts is a significant concrete approach to enhancing privacy assurance across autonomous security domains, and is considered a major contribution to the field of security and privacy.

References

[1] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard, "Web Services Architecture," W3C 2004.

[2] S. Anderson, J. Bohren, and e. al, "Web Services Trust Language (WS-Trust)," OASIS 2005.

[3] B. Carminati, E. Ferrari, and H. Patrick C.K, "Exploring Privacy Issues in Web Services Discovery Agencies," *IEEE Security and Privacy*, vol. 3, pp. 14-21, 2005.

[4] M.Lorch, S.Proctor, R.Lepro, D.Kafura, and S.Shah, " First Experience Using XACML for Access Control in Distributed Systems," presented at ACM Workshop on XML Security, Fairfax Va US,, 2003.

[5] E. J. C. R.S. Sandhu, H.L Feeinsten, C.E. Youman, "Role-Based Access Control Models," *IEEE Computer*, vol. Vol.29 (2), pp. 38-47, Feb 1996.

[6] G. Saunders, M. Hitchens, and V. Varadharajan, *An Analysis of Access Control Models*: Springer, 1999.

[7] O. O. D.W.Chadwick, "Implementing Role Based Access Controls Using X.509 Attribute Certificates," *IEEE Internet Computing*, pp. 62-69, 2003.

[8] E. Bertino, E.Ferrari, and A. Squicciarini, "Trust Negotiations: Concepts, Systems and Languages," *IEEE Computer*, pp. 27-34, 2004.

[9] N. Huntley, "Open and Closed Systems," http://www.users.globalnet.co.uk/~noelh/OpenClosedSystems.htm 2003.

[10] A.Acquisti, "Privacy and Security of Personal Information- Economics Incentives and Technological Solutions," presented at Workshop on Economics and Information Security, University of California Berkeley, 2002.

[11] K. E. Seamons, M.Winslett, T. Yu, L.Yu, and R.Jarvis, "Protecting Privacy during On-line Trust Negotiation,," presented at 2nd Workshop on Privacy Enhancing Technologies, San Francisco, CA, 2002.

[12] B. J.-B. Vanja Senicar, Tomaz Klobucar, "Privacy-Enhancing Technologies-approaches and development" Computer Standard & Interfaces," *Computer Standard & Interfaces 25*, pp. 147-158, 2003.

[13] OECD, "OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data," http://www.oecd.org/home/ 1980.

[14] FIP, "Privacy Online: Fair Information Practices in the Electronic Marketplace," http://www.ftc.gov/reports/privacy2000/privacy2000.pdf.

[15] CDT, "Center for Democracy and Technology : Privacy Policy, http://www.cdt.org/," http://www.cdt.org/ 2007.

[16] W3C, "Platform for Privacy Preferences (P3P)," 2004.

[17] OASIS, "Privacy policy profile of XACML," OASIS http://docs.oasis-open.org/ September 2004.

[18] S. Cantor, "Shibboleth Architecture," Internet2 Middleware http://shibboleth.internet2.edu/shibboleth-documents.html 2005.

[19] A. Acquisti and J. Grossklags, *What Can Behavioral Economics Teach Us About Privacy*: Auerbach Publications, 2007.

[20] A. Acquisti and J. Grossklags, "Privacy and Rationality in Individual Decision Making," *IEEE Security and Privacy*, vol. 3, pp. 26-33, 2005.

[21] J. W. E. Adkinson, J. A. Eisenach, and T. M. Lenard, "Privacy Online: A Report on the Information Practices and Polices of Commercial Websites," Progress and Freedom Foundation, http://www.pff.org/ September 2002.

[22] J. Linn, "Technology and Web User Data Privacy," *IEEE Security and Privacy*, vol. 3, pp. 52-58, 2005.

[23] A. Acquisti, "Privacy and Security of Personal Information- Economics Incentives and Technological Solutions," presented at Workshop on Economics and Information Security, University of California Berkeley, 2002.

[24] S. H. Paul Ashley, Gunter Karjoth , Calvin Powers, and Mathias Schunter, "Enterprise Privacy Authorization Language (EPAL 1.2)," presented at W3C Member Submission, 2003.

[25] S. Byers, L. Cranor, D. Kormann, and P. McDaniel, " Searching for Privacy: Design and Implementation of a P3P-Enabled Search Engine," presented at Workshop on Privacy Enhancing Technologies, Toronto, Canada, 2004.

[26] OPA, "Guidelines for Online Privacy Policies," Online Privacy Alliance.

[27] M. Hansen and J. Schallaböck, "Extending Policy Negotiation in User-Controlled Identity Management by Privacy & Security Information Services," presented at W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement, 2006.

[28] EU, "Directive 2002/58/EC on Privacy and Electronic Communications," European Parliament and the Council July 2002.

[29] C. W. Thompson and D. R. Thompson, "Identity Management," *IEEE Internet Computing*, vol. 11, pp. 82-85, 2007.

[30] U. Mbanaso, G. Cooper, D. Chadwick, and S. Proctor, "Privacy Preserving Trust Authorization using XACML," presented at Second International Workshop on Trust, Security and Privacy for Ubiquitous Computing (TSPUC 2006) Niagara-Falls, Buffalo-NY, 2006.

[31] U. M. Mbanaso, G. S. Cooper, D. Chadwick, and A. Anderson, "Obligations for Privacy and Confidentiality in Distributed Transactions," presented at Emerging Directions in Embedded and Ubiquitous Computing, Dec 2007, pp 69-81., 2007.

[32] K.E.Seamons, M.Winslett, T.Yu, B.Smith, E.Child, J.Jacobson, H.Mils, and L.Yu, "Requirements for Policy Languages for Trust Negotiation," presented at 3rd International Workshop on Policies for Distributed Systems and Networks, Moneterey, CA, 2002.

[33] M.Winslett, "An Introduction to Automated Trust Establishment," presented at 1st International Conference on Trust Management, Crete, Greece, 2003.

[34] S. Fisher-Hubner, *Lecture Notes in Computer Science: IT-Security and Privacy*, vol. 1958: Springer, 2001.

[35] S. H. Paul Ashley, Gunter Karjoth, "E-P3P Privacy Policies and Privacy Authorisation," presented at In Proceeding of the ACM workshop on Privacy in the Electronic Society, 2002.

[36] A. Anderson, "Comparison of Two Privacy Policy Languages: EPAL and XACML," in *5th Annual Privacy & Security Workshop*, vol. 2005: Sun Miicrosystems, 2005.

[37] T. Moses, "eXtensible Access Control Markup Language (XACML) Version 2.0," vol. 2005: OASIS, 2005.

[38] A. Anderson and H. Lockhart, "SAML 2.0 profile of XACML v2.0," OASIS February 2005.

[39] S. N. Sunder, K. Jamal, and M. S. Maier, "Privacy in e-Commerce: Development of Reporting Standards, Disclosure and Assurance Services in an Unregulated Market," *Journal of Accounting Research*, vol. 41, pp. 285-309, 2003.

[40] MetropolitanPolice, "Fraud Alert - Identity Theft/Fraud," Metropolitan Police: http://www.met.police.uk/fraudalert/section/identity_fraud.htm.

[41] CIFAS, "Is Identity Theft Serious?," CIFAS :http://www.cifas.org.uk/default.asp?edit_id=556-56.

[42] TIMESONLINE, "What is identity fraud," TIMESONLINE: http://business.timesonline.co.uk/tol/business/money/credit_clinic/article406173.ec e.

[43] S. Brostoff, M. A. Sasse, D. Chadwick, J. Cunningham, U. Mbanaso, and S. Otenko, "R-What? Development of a Role-Based Access Control (RBAC) Policy-Writing Tool for e-Scientists.," *Software: Practice and Experience* vol. 35(9), pp. 835-856, 2005.

[44] S. Brostoff, M. A. Sassea, D. Chadwick, J. Cunningham, U. Mbanaso, and O. Otenko, "RBAC what? Development of a role-based access control policy writing tool for e-Scientists," presented at Workshop on Grid Security Practice and Experience, Oxford UK, 2004.

[45] U. M. Mbanaso, "Privacy Preservation Architecture for Authorization Infrastructure," presented at 1RIS PhD Workshop 2005, 2005.

[46] U. M. Mbanaso, G. S. Cooper, Y. Rezgui, M. Wetherill, and S. C. Boddy, "Secure dynamic web services composition in the context of construction e-purchasing," presented at 24th W78 Conference Maribor 2007 & 5th ITCEDU Workshop & 14th EG-ICE Workshop, Maribor, Slovenia, 2006.

[47] U. M. Mbanaso, G. S. Cooper, D. Chadwick, and A. Anderson, "Obligations of Trust for Privacy and Confidentiality in Distributed Transactions," *INTERNET RESEARCH journal, Emerald. Special Issue: Intelligent Ubiquitous Computing: Applications and Security Issues*, vol. 19, pp. 153-172, 2009.

[48] M. Hansen, A. Schwartz, and A. Cooper, "Privacy and Identity Management," *IEEE Security & Privacy*, vol. 6, pp. 38-45, 2008.

[49] J. Creswell, *Qualitative inquiry and research design: Choosing among five traditions* Thousand Oaks, California: Sage Sage Publications, 1998.

[50] J. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Thousand Oaks, California: Sage Publications, 2003.

[51] B. J. Oates, *Researching Information Systems and Computing*: SAGE Publications Ltd.

[52] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, pp. 75-105, 2004.

[53] D. Straub, D. Gefen, and M.-C. Boudreau, "Quantitative, Positivist Research Methods in Information Systems," Association of Information Systems, http://dstraub.cis.gsu.edu:88/quant/: 2005.

[54] A. S. Lee, "Integrating Positivist and Interpretive Approaches to Organizational Research," *Organization Science*, pp. 342-365.

[55] W. J. Orlikowski and J. J. Baroudi, "Studying Information Technology in Organizations: Research Approaches and Assumptions," *Information Systems Research* pp. 1-28, 1991.

[56] C. W. Dawson, *Projects in Computing and Information Systems*: Pearson Education Limited, 2005.

[57] R. Vidgen, D. Avison, B. Wood, and T. Wood-Harper, *Developing Web Information Systems*: Butterworth-heinemann Information, 2003.

[58] P. Clough and C. Nutbrown, *A student's Guide to Methodology*, 1 ed: SAGE Publications Company, 2002.

[59] D. Andert, R. Wakefield, and J. Weise, "Trust Modeling for Security Architecture Development," Sun BluePrints OnLine, http://www.sun.com/blueprints/1202/817-0775.pdf December 2002.

[60] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual* 2nd Edition ed: The Addison-Wesley Object Technology Series.

[61] D. Chaum, " Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, pp. 1030 -2044, 1985.

[62] T. Scavo and S. Cantor, "Shibboleth Architecture," Shibboleth : http://www.cs.dartmouth.edu/~sws/papers/nazareth04.pdf, 2005.

[63] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role Based Access Control Models
" *IEEE Computer*, vol. 29, pp. 38-43, 1996.

[64] ISO, "ISO/IEC 10181-3:1996 Access Control Framework," International Organization for Standardization 1996.

[65] I. Sommerville, *Software Engineering*, 4 ed: Addison-Wesley, 1992.

[66] C. Floyd, *A Systematic Look at Prototyping*, vol. 1984: Springer Verlag

[67] Netbeans, "Integrated Development Environment " http://www.netbeans.org/ 2007.

[68] A. I. A. Qingfeng He, "A Framework for Modeling Privacy Requirements in Role Engineering,," in *Ninth International Workshop on Requirements Engineering: Foundation for Software Quality*. Klagenfurt/Velden, Austria, 2003.

[69] L. Kegal, T. Finin, A. Joshi, and S. Greenspan, "Security and Privacy Challenges in open and Dynamic Environments," *IEEE Computer Society*, pp. 89-91, 2006.

[70] R. E. Litan and A. M. Rivlin, "Projecting the Economic Impact of the Internet," *The American Economic Review, Papers and Proceedings of the Hundred Thirteenth Annual Meeting of the American Economic Association*, vol. 91, pp. 313-317, 2001.

[71] P. kuper, "The State of Security," *IEEE Security and Privacy*, vol. 3, pp. 50-53, 2005.

[72] S. D. Warren and L. D. Brandeis, "The Right To Privacy," *Harvard Law Review*, vol. 5, pp. 193-220.

[73] A. F. Westin, *Privacy and Freedom*: The Bodley Head Ltd 1970.

[74] A. Acquisti, "Identity Management, Privacy and Price Discrimination," *IEEE Security & Privacy*, vol. 6, pp. 46-50, 2008.

[75] E. Maler and D. Reed, "The venn Of Identity: Options and Issues in Federated Identity Management," *IEEE Security & Privacy*, vol. 6, pp. 16-21, 2008.

[76] P. N. Otto, A. I.Anton, and D. L. Baumer, "The ChoicePoint Dilemma: How Data Brokers Should Handle the Privacy of Personal Information," *IEEE Security and Privacy*, vol. 5, pp. 15-23, 2007.

[77] V. C. S. Lee and L. Shad, "Estimating Potential IT Security Losses," *IEEE Security and Privacy*, vol. 4, pp. 44-52, 2006.

[78] FTC, "Consumer Fraud and Identity Theft Compliant Data," US Federal Trade Commission : http://consumer.gov/sentinel/pubs/Top1oFraud2005.pdf 2006.

[79] ITRC, "Idenity Theft:The Aftermath 2004," Identity Theft Resource Center: http://idtheftcenter.org/aftermath2004.pdf Septemeber 2005.

[80] L. Cranor, J. Reagle, and M. Ackerman, "Beyond Concern: Understanding Net Users' Attitudes about Online Privacy " AT &T Research: www.reserach.att.com/projects/privacystudy 1999.

[81] BBC, "Sumitomo Mitsui bank fraud," http://news.bbc.co.uk/2/hi/technology/4358287.stm 2005.

[82] EPIC, "LexisNexis Breach Compromises Data on 310,000," 2005.

[83] A. Pfitzmann and e. el, "Anonymity, Unobservability, and Pesudonymity: A Proposal for Terminology," http://www.freehaven.net/anonbib/papers/Anon_Terminology_v0.14.pdf, 2003.

[84] B. Schneier, *Secrets and Lies-DigitalSecurity in a Networked World*, vol. 2: Wiley Publishing, Inc, 2004.

[85] D. Rowland, "Anonymity, Privacy and Cyberspace," presented at 5th BILETA Conference: Electronic Datasets and Access to Legal Information, University of Warwick, Coventry England, 2000.

[86] Liberty-Alliance, "Liberty Alliance ID-WSF Specifications," Liberty Alliance.

[87] J. D. Meier, "Web Application Security Engineering," *IEEE Security and Privacy*, vol. 4, pp. 16-24, 2006.

[88] K. M. Christopherson, "The positive and negative implications of anonymity in Internet social interactions: 'on the internet, nobody knows you're a dog'," *Computers in Human Behavior*, vol. 23, pp. 3038-56, 2007.

[89] FTC, "FEDERAL TRADE COMMISSION'S PRIVACY ONLINE:FAIR INFORMATION PRACTICES IN THE ELECTRONIC MARKETPLACE," http://www.ftc.gov/reports/ privacy2000/privacy2000text.pdf, May 2000.

[90] J. Jonas, "Threat and Fraud Intelligence, Las Vegas Style," *IEEE Security and Privacy*, vol. 4, pp. 28-34, 2006.

[91] G.-L. B. Act, "Financial Privacy and Pretexting," *Federal Trade Commission*, http://www.ftc.gov/privacy/glbact/index.html.

[92] APEC, "APEC Privacy Framework," Asia-Pacific Economic Cooperation(APEC): http://www.apec.org 2005.

[93] P. Cole, W. Duerick, J. Lasser, G. Podorowsky, P. Sibieta, and C. Thornby, "Privacy and Security Best Practices," Liberty Alliance Project 2003.

[94] D.W.Chadwick and D.P.Mundy, "Policy Based Electronic Transmission of Prescriptions," presented at IEEE 4th International Workshop on Policies for Distributed Systems and Networks, Como Italy, 2003.

[95] D.W.Chadwick and O.Otenko, "Implementing Role Based Access Controls Using X.509 Attribute Certificates," *IEEE Internet Computing*, pp. 62-69, 2003.

[96] C. Adams, "Building Secure Web-Based Environments," *IEEE Security and Privacy*, vol. 3, pp. 74-77, 2005.

[97] Q. He, "Privacy Enforcement with Extended Role-Based Access Control Model," *NCSU Computer Science Technical Report TR-2003-09*, 2003.

[98] A. T. Vassilev, B. D. Castel, and A. M. Ali, "Personal Brokerage of Web Services Access," *IEEE Security and Privacy*, vol. 5, pp. 24-31, 2007.

[99] J. Tourzan and Y. Koga, "Liberty ID-WSF Web Services Framework Overview," Liberty Alliance.

[100] I. I.-. ITU-T Recommendation X.509, "Information Technology - Open Systems Interconnection " ITU.

[101] J. Kohl and B. C. Neuman, "The Kerberos Network Authentication Service," September 1993.

[102] B. Schneier, *Applied Cryptography*, Second Edition ed: John Wiley & Sons, 1996.

[103] J. G. Steiner, B. C. Neuman, and J. I. Schiller, " Kerberos: An Authentication Service for Open Network Systems," presented at Winter 1988 Usenix Conference, 1998.

[104] D. Mazurek, "Central Authentication Service Architecture Protocol," Yale University 2005.

[105] OpenID, "The OpenID " http://openid.net/what/ 2006.

[106] D. J. Weitzner, "Whose Name Is It, Anyway?," *IEEE Internet Computing*, pp. 72-76, 2007.

[107] S. Cantor, J. Kemp, R. Philpott, and E. Maler, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) Version 2.0," vol. 2005: OASIS, 2005.

[108] D.W.Chadwick, "The X.509 Privilege Management Infrastructure," presented at Proceedings of the NATO Advanced Networking Workshop on Advanced Security Technologies in Networking, Bled, Slovenia, 2003.

[109] O. Otenko, "Policy-Based Privilege Management Using X.509," in *Information Systems Research Institute.* Salford: University of Salford, 2004, pp. 252.

[110] D. Chadwick, "The X.509 Privilege Management Infrastructure," presented at NATO Advanced Networkign Workshop on Advanced Security Technologies in Networking, Bled, Slovenia, 2003.

[111] D. W. Chadwick and A. Otenko, "RBAC Policies in XML for X.509 Based Privilege Management," presented at Security in the Information Society: Visions and Perspectives: IFIP TC11 17th Int. Conf. On
Information Security (SEC2002), May 7-9, 2002, Cairo, Egypt. Ed. by, Cairo, Egypt, 2002.

[112] D.W.Chadwick and D.P.Mundy, "The Secure Electronic Transfer of Prescriptions," presented at HC2004, , Harrogate, UK, 2004.

[113] D. Chadwick, S. Otenko, Wensheng, and W. Xu, "Adding Distributed Trust Management to Shibboleth," presented at NIST 4th Annual PKI Workshop, Gaithersherg, USA, 2005.

[114] W. Xu, D. Chadwick, and S. Otenko, "Development of a Flexible PERMIS Authorisation Module for Shibboleth and Apache Server," presented at 2nd EuroPKI Workshop, University of Kent, 2005.

[115] M. Blaze, J. Feigenbaum, and A. D. Keromytis, "KeyNote: Trust management for publickey infrastructures " presented at Security Protocols International Workshop, Cambridge, 1998.

[116] K. E. Seamons, T. Ryutov, L. Zhou, C. Neuman, and T. Leithead, "Adaptive Trust Negotiation and Access Control," presented at 10th ACM Symposium on Access Control Models and Technologies, ,Stockholm, Sweden, 2005.

[117] W. H. Winsborough and N. Li, "Towards Practical Automated Trust Negotiation," presented at Proceedings of the Third International Workshop on Policies for
Distributed Systems and Networks (Policy 2002), 2002.

[118] J.Holt and K.E.Seamons, "Interoperable Strategies in Automated Trust Negotiation," presented at 8th ACM Conference on Computer and Communications Security, Philadelphia Pennsylvania, 2001.

[119] T.Barlow, A.Hess, and K.E.Seamons, "Trust Negotiation in Electronic Markets.," presented at Eighth Research Symposium in Emerging Electronic Markets, Maastrict Netherlands, 2001.

[120] T. Ryutov, L. Zhou, C. Neuman, N. Foukia, T. Leithead, and K. E. Seamons, "Adaptive Trust Negotiation and Access Control for Grids," presented at 6th IEEE/ACM International Workshop on Grid Computing, Seattle, WA, 2005.

[121] E. F. E.Bertino, A Squicciarini, "TNL: An XML-based Language for Trust Negotiations," presented at IEEE 4th International Workshop on policies for Distributed Systems and Networks, Lake Como Italy, 2003.

[122] E. F. E.Bertino and A. Squicciarini, "TNL: An XML-based Language for Trust Negotiations," presented at IEEE 4th International Workshop on policies for Distributed Systems and Networks, Lake Como Italy, 2003.

[123] K. E. Seamons, M. Winslett, and T. Yu, " Limiting the Disclosure of Access Control Policies During Automated Trust Negotiation," presented at Network and Distributed System Security Symposium, San Diego, CA, 2001.

[124] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, "The KeyNote Trust-Management System Version 2," Network Working Group, RFC September 1999.

[125] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized Trust Management. ," presented at 17th IEEE Symp. on Security and Privacy., 1996.

[126] E. Bertino, E. Ferrari, and A. Cinzia, "Trust - X: A Peer-to-Peer Framework for Trust Establishment," *IEEE Transactions on Knowledge and Data Engineering* vol. 16, pp. 827-849, 2004.

[127] C. C. Morris, "Browser-Based Trust Negotiation," in *Department of Computer science*: Brigham Young University, 2006.

[128] W. Hommel, "Using XACML for Privacy Control in SAML-Based Identity Federations," *CMS 2005, LNCS 3677(IFIP International Federation for Information Processing 2005)*, pp. 160–169, 2005.

[129] A. Ekert, C. M. Alves, and A. Gopinathan, "History of Cryptography," University of Cambridge.

[130] D. R. Stinson, *Cryptography Theory and Practice*: CRC Press Inc. USA, 1995.

[131] C. Adams and S. Lloyd, *Understanding PKI: Concepts, Standards and Deployment Considerations*, 2 ed: Addison-Wesley, 1999.

[132] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, pp. 644-654.

[133] http://www.cypher.com.au/crypto_history.htm, "A Brief History of Cryptography."

[134] B. Forouzan, *Introduction to Cryptography and Network Security*: MeGraw-Hill Higher Education, 2008.

[135] H. Deffs and H. Knebl, *Introduction to Cryptography Principles and Applications*: Springer, 2002.

[136] NIST, "Digital Signature Standard, Federal Information Processing Standards Publication 186," U.S. Department of Commerce, National Bureau of Standards, Springfield VA 1994.

[137] NIST, "Secure Hash Standard, Federal Information Processing Standards Publication 180-1," U.S. Department of Commerce, National Bureau of Standards, Springfield, VA 1995.

[138] I.-T. R. X.509, "Information Technology - Open Systems Interconnection- The Directory: Public Key and Attribute Certificate Frameworks," ITU March 2000.

[139] RFC2538, "RFC2538: Storing Certificates in the Domain Name System," http://rfc.net/rfc2538.html March 1999.

[140] IETF, "RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile," IETF available at http://www.ietf.org/rfc/rfc2459.txt 1999.

[141] RFC2693, "SPKI Certificate Theory," http://www.ietf.org/rfc/rfc2693.txt 1999.

[142] RFC2692, "SPKI Requirements," http://www.ietf.org/mail-archive/web-old/ietf-announce-old/current/msg05215.html 1999.

[143] RFC4880, "OpenPGP Message Format," IETF:http://www.ietf.org/rfc/rfc4880.txt 2007.

[144] RFC3280, " Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, http://www.ietf.org/rfc/rfc3280.txt," IETF 2002.

[145] IETF, "IETF Standards," IETF, http://www.ietf.org/.

[146] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*: Pearson Education Addison-Wesley, 2001.

[147] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible Markup Language (XML) 1.0 " W3C : http://www.w3.org/TR/REC-xml/ August 2006.

[148] W3C, "The Platform for Privacy Preferences 1.0 (P3P 1.0)," Technical Report. 2002.

[149] A. Anderson, "Web Services Profile of XACML (WS-XACML) Version 1.0, WD 8," OASIS XACML Technical Committee 12 December 2006.

[150] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, "Enterprise Privacy Authorization Language (EPAL 1.2)," IBM, http://www.zurich.ibm.com/security/enterprise-privacy/epal, 2003.

[151] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, "Enterprise Privacy Authorization Language," IBM, http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/ November 2003.

[152] OASIS, "XACML Context Schema," http://www.oasis-open.org/committees/xacml/repository/cs-xacml-schema-context-01.xsd 2005.

[153] S. Cantor, J. Kemp, R. Philpott, and E. Maler, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0," OASIS:http://www.oasis-open.org/specs/index.php#samlv2.0, 15 March 2005.

[154] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker, "Web Services Security," Organization for the Advancement of Structured 1information Standards(OASIS) February 2006.

[155] P. Cole, P. Lord, J. Alhadeff, R. Wilton, J. Winn, and C. Allis, "Liberty Alliance Contractual Framework Outline For Circle of Trust," Liberty Alliance Project.

[156] S. Johnson and R. Sethi, *Yacc:A parser generator*, vol. 2: Saunders College Publishing, philadelphia, Pennsylvania, 1990.

[157] RecommendationX.208, " Open Systems Interconnection:Specification of Abstract Syntax Notation (ASN.1)."

[158] K. E. Seamons, M. Winslett, Y. Ting, B. Smith, E.Child, J. Jacobson, H. Mills, and Y. Lina, "Requirements for policy languages for trust negotiation," presented at Third International Workshop on Policies for Distributed Systems and Networks 2002.

[159] D. W. Chadwick, S. Otenko, and T. A. Nguyen, "Adding Support to XACML for Dynamic Delegation of Authority in Multiple Domains," presented at 10th IFIP TC-6 TC-11 Int Conf, CMS 2006, Heraklion, Crete, Greece, 2006.

[160] Y. Demchenko, "Using XACML and SAML for Authorisation messaging and assertions:," vol. 2005, 2005.

[161] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker, "Web Services Security: SOAP Message Security 1.1," OASIS:http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf 2006.

[162] W. H. Winsborough, K. E. Seamons, and V. E. Jones, " Automated trust negotiation.," presented at Information Survivability Conference and Exposition, 2000.

[163] V. E. Jones, W. H. Winsborough, and K. Seamons, "TRUST MANAGEMENT IN OPEN SYSTEMS (TMOS)," http://stinet.dtic.mil/cgi-bin/GetTRDoc?AD=ADA403753&Location=U2&doc=GetTRDoc.pdf.

[164] A. Anderson and U. Mbanaso, "Schema for Obligation of Trust (OoT)," IRIS, University of Salford, http://infosec.salford.ac.uk/names/oot/ootSchema/ December 2006.

[165] A. Anderson, "Schema for Web Services Profile of XACML (WS-XACML) Version 1.0, WD 8, 2006," OASIS XACML Technical Committee, 12 December 2006.

[166] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns, Elements of Reusable Object-Oriented Software*: Addison-Wesley, 1995.

[167] M. Grand, *Patterns in Java*, vol. 1, second ed: Wiley Publishing, Inc, 2002.

[168] S. Proctor, "Sun XACML implementation APIs," vol. 2005: http://sunxacml.sourceforge.net/, 2004.

[169] S. Microsystems, "ava Platform, Enterprise Edition (Java EE)," Sun Microsystems http://java.sun.com/javaee/.

[170] JAX-WS: https://jax-ws.dev.java.net/.

[171] SUN, "JAX-WS API : http://java.sun.com/mailers/techtips/enterprise/2006/TechTips_June06.html."

[172] W. C. Hetzel, *The Complete Guide to Software Testing*: Wellesley, Mass. : QED Information Sciences, 1988.

[173] C. U. Smith and L.G.Williams, *Performance Evaluation: a practical guide to creating responsive, sclable software*: Addision Wesley, 2002.

[174] Shibboleth, "Link of Shibboleth Deployment," http://switch.ch/aai/participants/.

[175] K. Martin, "Privacy and anonymity," Security Focus February 2006.

[176] EU, "REPORT FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT AND THE COUNCIL Report on the operation of Directive 1999/93/EC on a Community framework for electronic signatures," EU, Brussels March 2006.

Appendix A: SAML Obligation of Trust Schema

The code below is the XML schema document which describes the structure of the SAML Obligation of Trust protocol request-response messages, discussed in chapter 8. Obligation of Trust protocol is an extension of the SAML request-response protocol discussed in chapter 6.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  targetNamespace="http://infosec.salford.ac.uk/names/oot/"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:oot="http://infosec.salford.ac.uk/names/oot/"
  xmlns:ws-xacml="urn:oasis:names:tc:xacml:3.0:profile:webservices:v1.0:schema"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified"
  blockDefault="substitution"
  version="WD 1">
  <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
    schemaLocation="http://docs.oasis-open.org/security/saml/v2.0/saml-schema-assertion-2.0.xsd"/>
  <import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
    schemaLocation="http://docs.oasis-open.org/security/saml/v2.0/saml-schema-protocol-2.0.xsd"/>
  <import namespace="urn:oasis:names:tc:xacml:3.0:profile:webservices:v1.0:schema"
    schemaLocation="http://www.oasis-open.org/committees/download.php/21411/xacml-3.0-profile-webservices-
schema-v1.0-wd-7.xsd"/>
  <annotation>
   <!-- The Target Namespace, Document identifier and Location are provisional -->
   <documentation>
     Document identifier: oot-wd-1
     Location: http://
   </documentation>
  </annotation>
  <!--  -->
  <element name="ObligationOfTrustQuery" xsi:type="oot:ObligationOfTrustQueryType" />
  <complexType name="ObligationOfTrustQueryType">
    <complexContent>
      <extension base="samlp:RequestAbstractType">
        <sequence>
          <element ref="ws-xacml:XACMLPrivacyAssertion" minOccurs="1" maxOccurs="unbounded" />
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!--  -->
  <complexType name="ObligationOfTrustStatementType">
    <complexContent>
      <extension base="saml:StatementAbstractType">
        <sequence>
          <element ref="ws-xacml:XACMLPrivacyAssertion" minOccurs="1" maxOccurs="1" />
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!--  -->
</schema>
```

Appendix B Matching Algorithms

The table below describes the various WS-XACML matching algorithms in the context of evaluating the OoT message contexts discussed in chapter 8.

Foreign XACMLAssertion Requirements	Local XACMLAssertion Capabilities	Match Algorithm
Missing *Requirements* element	Any	Always TRUE. The providing party of the foreign XACMLAssertion is not willing to provide *Requirements*, but is indicating that an XACML policy MAY be applied at the time of the interaction.
Any	Missing *Capabilities* element	Always TRUE. The providing party of the local XACMLAssertion is not willing to provide *Capabilities*, but is acknowledging that an XACML policy MAY be applied at the time of the interaction.
Empty *Requirements* element	Any	Always TRUE
Xacml:policy or xacml:policySet	Xacml-context:Request or xacml:ResourceContent	TRUE if the vocabularies match AND if there is at least one *Request* or *ResourceContent* element in the *Capabilities* that when evaluated against policy or PolicySet according to the XACML standard returns "Permit"; otherwise "False"
Xacml:Policy or xacml:policySet	Xacml:Apply	Not defined
Xacml:Apply	Xacml-context:Request or Xacml-context:ResourceContent	Consider the *Requirements* to be semantically equivalent to the XACML policy by the *Requirements* spec. TRUE if the vocabularies match AND if there is at least one *Request* or *ResourceContent* element in the *Capabilities* that when evaluated against the Policy according to

		the XACML standard returns "Permit"; otherwise "False".
Xacml:Apply	XACML:Apply	TRUE if the vocabularies match AND if, for each Apply element in the *Requirements*, there is at least one *Apply* element in the *Capabilities* for which the intersection defined in Appendix A in [165] is non-empty; otherwise "FALSE".

Appendix C Test Plan

The table below provides a complete description of the various test-cases used in the technical validation discussed in chapter 9, and shows the purpose of each test, actual test carried out and an expected result.

Test	**PURPOSE**	**TEST CARRIED OUT**	**EXPCTED RESULT**
Test-case 1	The test seeks to test the capability of the system to restrict access to protected resources to guarantee that only users with the right tokens are authorized to access the resources. In this test privacy is not a concern, the client submits the token issued by its domain STS because the token does not explicitly expose client's PII.	The test is performed using a set of data representing *request context* and *resource policy* as inputs. Two tests will be carried out in which one test should have the appropriate subject descriptor attribute and the other without.	It is expected that the request with the required subject descriptor attribute will be allowed, and the other denied.
Test-case 2	The test seeks to evaluate the ability of the system to respond to access requests with insufficient information in the subject descriptor. This test aims to achieve two purposes. From the client's perspective, it is not ready to supply all the information required in one got without determining first how the recipient is going to treat the information. From the service provider side, it is not prepared to advertise full access rules to arbitrary strangers. So both parties want to ensure privacy and confidentiality using the trust negotiation.	The test is performed using a set of policies between the client and service, and *request context* as inputs. The service has a local resource policy and a privacy assertion policy. The client has the request context and privacy assertion policy. In the test, the two parties performed privacy negotiation, advertised their *Requirements* and *Capabilities,* and the systems performed matching and evaluation with corresponding results. This test demonstrates the parsimonious strategy in which parties are not willing to refine their *Requirements* or *Capabilities* during negotiation.	It is expected that both parties can meet each other's *Requirements* by a set of *Capabilities* to make the negotiation succeed without too many iterations. The successful negotiation will result in issuance of SAO by both parties as a guarantee that they have agreed to respect each other's privacy.

Test-case 3	The test is similar to test-case 2 but differs in the negotiation strategy. This is suitable where parties may not want to advertise all their Requirements and Capabilities, but would expose their privacy assertions incrementally and gradually to advance the threshold of trust. This is equally useful for mitigating against probing or inferential attacks.	This test demonstrates the persuasive strategy in which parties are willing to refine their *Requirements* or *Capabilities* during negotiation wherever there are mismatches in their privacy assertions.	It is expected that one party's *Requirements* will not be met by the other party's set of *Capabilities*, so that this party sends a counter NoB to advertise its *Requirements* and *Capabilities* to enable the party to refine its *Capabilities* and if appropriate *Requirements* to advance the negotiation success. The successful negotiation will result in issuance of SAO by both parties as a guarantee that they have agreed to respect each other's privacy.
Test-case 4 (Performance test)	This test seeks to measure the throughput in terms of average response time in the full cycle of an access control session including the privacy negotiation between two parties.	A set of data representing *request context* and policies were used as inputs to carry out a range of tests. The different datasets represent workloads on the systems In order to measure different response times.	It is expected that different datasets will give different response times. The response time depends on a number of factors including the volume of the input dataset to be processed by the system.

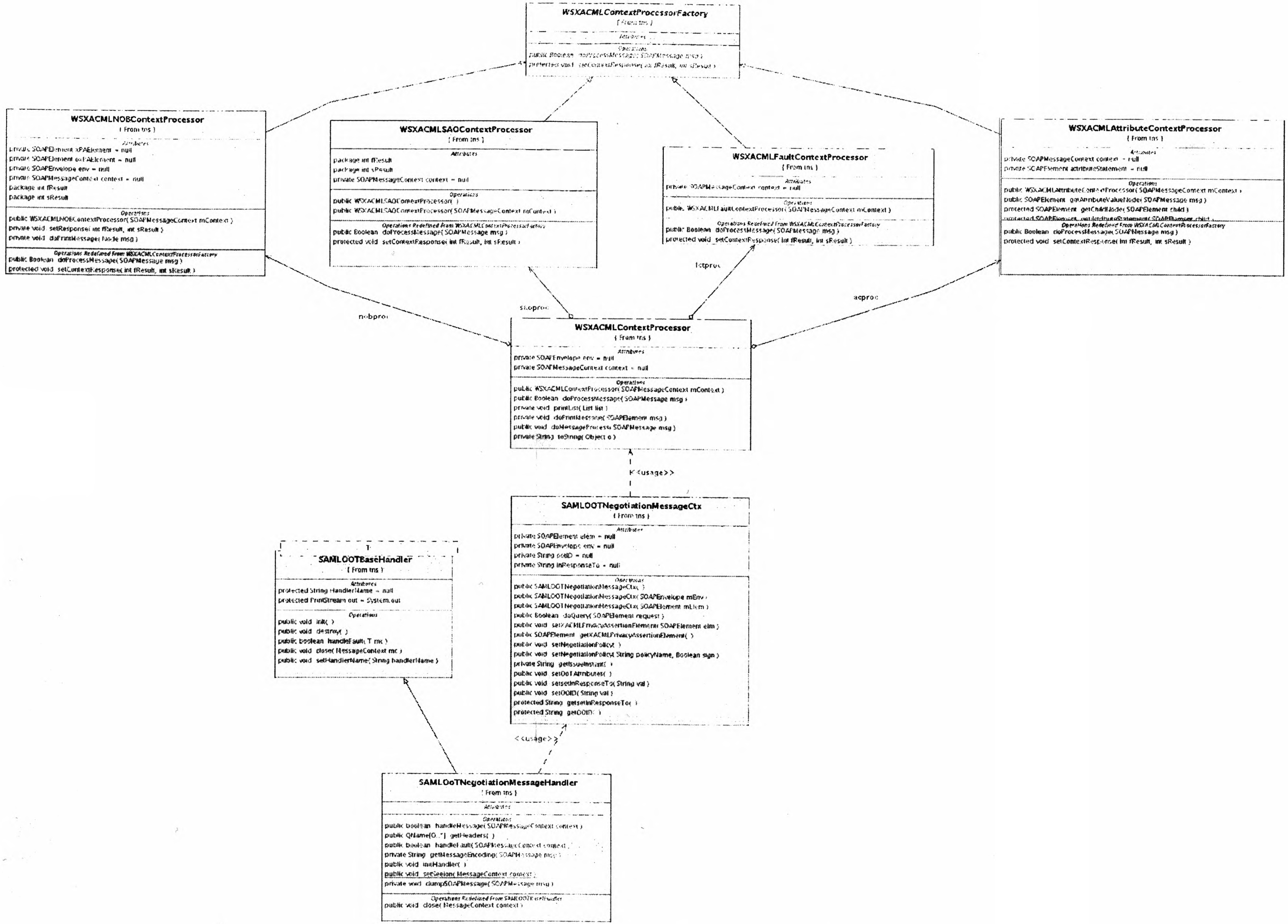

WSXACMLContextProcessorFactory
{ From tns }
Attributes
Operations
public Boolean doProcessMessage(SOAPMessage msg)
protected void setContextResponse(int fResult, int sResult)
WSXACMLNOBContextProcessor
{ From tns }
Attributes
private SOAPElement xPAElement = null
private SOAPElement oxPAElement = null
private SOAPEnvelope env = null
private SOAPMessageContext context = null
package int fResult
package int sResult
Operations
public WSXACMLNOBContextProcessor(SOAPMessageContext mContext)
private void setResponse(int fResult, int sResult)
private void doPrintMessage(Node msg)
Operations Redefined From WSXACMLContextProcessorFactory
public Boolean doProcessMessage(SOAPMessage msg)
protected void setContextResponse(int fResult, int sResult)
WSXACMLSAOContextProcessor
{ From tns }
Attributes
package int fResult
package int sResult
private SOAPMessageContext context = null
Operations
public WSXACMLSAOContextProcessor( )
public WSXACMLSAOContextProcessor(SOAPMessageContext mContext)
Operations Redefined From WSXACMLContextProcessorFactory
public Boolean doProcessMessage(SOAPMessage msg)
protected void setContextResponse(int fResult, int sResult)
WSXACMLFaultContextProcessor
{ From tns }
Attributes
private SOAPMessageContext context = null
Operations
public WSXACMLFaultContextProcessor(SOAPMessageContext mContext)
Operations Redefined From WSXACMLContextProcessorFactory
public Boolean doProcessMessage(SOAPMessage msg)
protected void setContextResponse(int fResult, int sResult)
WSXACMLAttributeContextProcessor
{ From tns }
Attributes
private SOAPMessageContext context = null
private SOAPElement attributeStatement = null
Operations
public WSXACMLAttributeContextProcessor(SOAPMessageContext mContext)
public SOAPElement getAttributeValueNode(SOAPMessage msg)
protected SOAPElement getChildNode(SOAPElement child)
protected SOAPElement getAttributeStatement(SOAPElement child)
Operations Redefined From WSXACMLContextProcessorFactory
public Boolean doProcessMessage(SOAPMessage msg)
protected void setContextResponse(int fResult, int sResult)
nobproc
saoproc
fctproc
acproc
WSXACMLContextProcessor
{ From tns }
Attributes
private SOAPEnvelope env = null
private SOAPMessageContext context = null
Operations
public WSXACMLContextProcessor(SOAPMessageContext mContext)
public Boolean doProcessMessage(SOAPMessage msg)
private void printList(List list)
private void doPrintMessage(SOAPElement msg)
public void doMessageProcess(SOAPMessage msg)
private String toString(Object o)
<<usage>>
SAMLOOTNegotiationMessageCtx
{ From tns }
Attributes
private SOAPElement elem = null
private SOAPEnvelope env = null
private String ootID = null
private String inResponseTo = null
Operations
public SAMLOOTNegotiationMessageCtx( )
public SAMLOOTNegotiationMessageCtx(SOAPEnvelope mEnv)
public SAMLOOTNegotiationMessageCtx(SOAPElement mElem)
public Boolean doQuery(SOAPElement request)
public void setXACMLPrivacyAssertionElement(SOAPElement elm)
public SOAPElement getXACMLPrivacyAssertionElement( )
public void setNegotiationPolicy( )
public void setNegotiationPolicy(String policyName, Boolean sign)
private String getIssueInstant( )
public void setOoTAttributes( )
public void setsetInResponseTo(String val)
public void setOOID(String val)
protected String getsetInResponseTo( )
protected String getOOID( )
T
SAMLOOTBaseHandler
{ From tns }
Attributes
protected String HandlerName = null
protected PrintStream out = System.out
Operations
public void init( )
public void destroy( )
public boolean handleFault(T mc)
public void close(MessageContext mc)
public void setHandlerName(String handlerName)
<<usage>>
SAMLOoTNegotiationMessageHandler
{ From tns }
Attributes
Operations
public boolean handleMessage(SOAPMessageContext context)
public QName[0..*] getHeaders( )
public boolean handleFault(SOAPMessageContext context)
private String getMessageEncoding(SOAPMessage msg)
public void initHandler( )
public void setSession(MessageContext context)
private void dumpSOAPMessage(SOAPMessage msg)
Operations Redefined From SAMLOOTBaseHandler
public void close(MessageContext context)

Figure Appendix D TNH Module Class Diagram discussed in Chapter 9

Appendix E: Test-Cases Transcripts

The complete transcripts of the various test-cases discussed in chapter 9 are documented in the following section.

E.1: Test case 1: XACML request contexts, and XACML response contexts discussed in chapter 9

```
<Request>
<Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"><AttributeValue>1214901122877@xtmai.sts.co.uk</AttributeValue></Attribute>
</Subject>
<Resource>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"><AttributeValue>http://xtmai.tcbportal.com/</AttributeValue>
</Attribute>
</Resource>
<Action>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>read</AttributeValue></Attribute>
</Action>
</Request>

<Response>
<Result ResourceId="http://xtmai.tcbportal.com/">
<Decision>Permit</Decision>
<Status>
<StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
</Status>
</Result>
</Response>
```

```
<Request>
 <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"><AttributeValue>1214901315128@pcg.org</AttributeValue></Attribute>
 </Subject>
 <Resource>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"><AttributeValue>http://xtmai.tcbportal.com/</AttributeValue>
</Attribute>
 </Resource>
 <Action>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"><AttributeValue>read</AttributeValue></Attribute>
 </Action>
</Request>

<Response>
 <Result ResourceId="http://xtmai.tcbportal.com/">
  <Decision>Deny</Decision>
  <Status>
   <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
  </Status>
 </Result>
</Response>
```

E. 1.2: XACML Local Policy used in test case 1

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      PolicyId="GeneratedPolicy"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.1:rule-combining-algorithm:ordered-permit-overrides">
  <Description>

  </Description>
  <Target>
     <Subjects>
       <AnySubject/>
     </Subjects>
     <Resources>
       <AnyResource/>
     </Resources>
    <Actions>
     <AnyAction/>
    </Actions>
    </Target>

 <Rule RuleId="CommitRule" Effect="Permit">
<Target>
    <Subjects>
     <Subject>
       <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">[redacted]</AttributeValue>
         <SubjectAttributeDesignator DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"
                             AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"/>
       </SubjectMatch>
       </Subject>
     <Subject>
      <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">[redacted]</AttributeValue>
         <SubjectAttributeDesignator DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"
                             AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"/>
       </SubjectMatch>
     </Subject>
   </Subjects>
   <Resources>
     <Resource>
       <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">[redacted]</AttributeValue>
         <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#anyURI"
                             AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
       </ResourceMatch>
     </Resource>
   </Resources>
   <Actions>
      <Action>
        <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">[redacted]</AttributeValue>
          <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                             AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
        </ActionMatch>
      </Action>
    </Actions>
 </Target>
  </Rule>
 <Rule RuleId="FinalRule" Effect="Deny"/>
</Policy>
```

E.2.1 NoB context sent to the client by the service described in test case 2 in chapter 9

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:oot="http://infosec.salford.ac.uk/names/oot"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope" xmlns:ws-
xacml="urn:oasis:names:tc:xacml:3.0:profile:webservices" xmlns:wsse="http://schema.xmlsoap.org/ws/2002/xx/sece"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <S:Header>
      <wsse:Security MustUnderstand="1">
        <oot:ObligationOfTrustQuery IssuerInstant="2008-07-01T09:04:16Z" ootID="1214903056554">
          <saml2:Assertion ID="1214903056554" IssueInstant="2008-07-01T09:04:16Z">
            <saml2:Issuer>http://xtmai.sts.salford.ac.uk/service</saml2:Issuer>
            <ws-xacml:XACMLPrivacyAssertion>
                <Requirements>
                  <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
                 <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
                     <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
                        <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
                               DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                     </Apply>
                    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">30</AttributeValue>
                 </Apply>
                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                    <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                     AttributeId="pcg-groups"
                                     Issuer="admin@pcg.org"/>
                  </Apply>
                </Requirements>
             <Capabilities>
                  <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
                  <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
                     <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
                        <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
                              DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                     </Apply>
                     <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">40</AttributeValue>
                  </Apply>
          </Capabilities>
            </ws-xacml:XACMLPrivacyAssertion>
          </saml2:Assertion>
        </oot:ObligationOfTrustQuery>
      </wsse:Security>
    </S:Header>
    <S:Body>
      <startNegotiation xmlns="http://provider.ws.xacml.xtmai.iris.salford.ac.uk/"/>
      <NegotiationState>
        <Service role="uk:ac:salford:iris:oot:service:role:client">
          <State value="uk:ac:salford:iris:oot:state:nob"/>
          <Status value="uk:ac:salford:iris:oot:status:ok"/>
        </Service>
      </NegotiationState>
    </S:Body>
  </S:Envelope>
<S.Envelope xmlns:S="http //schemas.xmlsoap org/soap/envelope/" xmlns:oot="http://infosec salford.ac.uk/names/oot"
```

E.2.2 SAO context sent to the service by the client described in test case 2 in chapter 9

```
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope" xmlns:ws-
xacml="urn:oasis:names:tc:xacml:3.0:profile:webservices" xmlns:wsse="http://schema.xmlsoap.org/ws/2002/xx/sece"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <S:Header>
      <wsse:Security MustUnderstand="1">
        <oot:ObligationOfTrustStatement InResponseTo="1214903056554" IssuerInstant="2008-07-01T09:04:16Z" ootID="1214903056590">
          <saml2:Assertion ID="1214903056590" IssueInstant="2008-07-01T09:04:16Z">
            <saml2:Issuer>http://xtmai.sts.salford.ac.uk/client</saml2:Issuer>
            <ws-xacml:XACMLPrivacyAssertion>
                  <Requirements>
                    <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
                    <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
                       <Apply AttributeId="urn:oasis:names:tc:xacml.1.0:function:integer-less-than-or-equal">
                         <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
                                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                       </Apply>
                    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">45</AttributeValue>
                    </Apply>
                  </Requirements>
                  <Capabilities>
                    <Vocabulary>urn:oasis:tc:xacml.3.0:vocabulary.xacml</Vocabulary>
                    <Apply AttributeId="urn:oasis:names:tc:xacml:3.0.function:must-be-present">
                      <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
                         <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action.max-data-rention-days"
                                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                      </Apply>
                      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">25</AttributeValue>
                    </Apply>
                  <Apply FunctionId="urn:oasis:names:tc.xacml:1.0:function:string-one-and-only">
                      <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                     AttributeId="pcg-groups"
                                     Issuer="admin@pcg.org"/>
                    </Apply>
                  </Capabilities>
            </ws-xacml:XACMLPrivacyAssertion>
            <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
              <SignedInfo>
                <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#WithComments"/>
                <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
                <Reference URI="">
                  <Transforms>
                    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                  </Transforms>
                  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <DigestValue>j/XO/Ajdq0EPyppMcmVGDp6eZRs=</DigestValue>
                </Reference>
              </SignedInfo>
              <SignatureValue>Z8o1Pbv7+103m+oXbfOdG2HnuRAuoZETJFQcVpWxvNeINgJ3DPPl7w==</SignatureValue>
              <KeyInfo>
                <KeyValue>
                  <DSAKeyValue>
                    <P>/KaCzo4Syrom78z3EQ5SbbB4sF7ey80etKII864WF64B81uRpH5t9jQTxeEu0ImbzRMqzVDZkVG9xD7nN1kuFw==
</P>
                    <Q>li7dzDacuo67Jg7mtqEm2TRuOMU=</Q>

<G>Z4Rxsnqc9E7pGknFFH2xqaryRPBaQ01khpMdLRQnG541Awtx/XPaF5Bpsy4pNWMOHCBiNU0NogpsQW5QvnlMpA== </G>
                    <Y>qcDjwk0fNNxItgis6yZvOb1RDtfZWvFjpLZX27zqbaZZxWh0cn17pD9fhyjV6XMHWSh8QgxyxbzyLXld+ZrXrw==
</Y>
                  </DSAKeyValue>
                </KeyValue>
              </KeyInfo>
            </Signature>
          </saml2:Assertion>
        </oot:ObligationOfTrustStatement>
      </wsse:Security>
    </S:Header>
    <S:Body>
      <NegotiationState xmlns="">
        <Service role="uk:ac:salford:iris:oot:service:role:provider">
          <State value="uk:ac:salford:iris:oot:state:sao"/>
          <Status value="uk:ac:salford:iris:oot:status:ok"/>
        </Service>
      </NegotiationState>
    </S:Body>
  </S:Envelope>
```

E.2.3 SAO context sent to the client by the service described in test case 2 in chapter 9

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:oot="http://infosec.salford.ac.uk/names/oot"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope" xmlns:ws-
xacml="urn:oasis:names:tc:xacml:3.0:profile:webservices" xmlns:wsse="http://schema.xmlsoap.org/ws/2002/xx/sece"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <S:Header>
      <wsse:Security MustUnderstand="1">
        <oot:ObligationOfTrustStatement InResponseTo="1214903056590" IssuerInstant="2008-07-01T09:18:28Z" ootID="1214903908126">
          <saml2:Assertion ID="1214903908139" IssueInstant="2008-07-01T09:18:28Z">
            <saml2:Issuer>http://xtmai.sts.salford.ac.uk/service</saml2:Issuer>
            <ws-xacml:XACMLPrivacyAssertion>
              <Requirements>
                <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
                <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
                  <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
                    <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                  </Apply>
                  <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">30
                  </AttributeValue>
                </Apply>
              </Requirements>
              <Capabilities>
                <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
                <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
                  <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
                    <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                  </Apply>
                  <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">25
                  </AttributeValue>
                </Apply>
              </Capabilities>
            </ws-xacml:XACMLPrivacyAssertion>
            <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
              <SignedInfo>
                <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#WithComments"/>
                <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
                <Reference URI="">
                  <Transforms>
                    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                  </Transforms>
                  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <DigestValue>SHWlLpElngchIW4tFI9n8UguHU4=</DigestValue>
                </Reference>
              </SignedInfo>
              <SignatureValue>a0zn0ADt2exarn0Xj0ELMKPi6J1WHntlaqneDh3P62qegnq57qHDwg==</SignatureValue>
              <KeyInfo>
                <KeyValue>
                  <DSAKeyValue>

<P>/KaCzo4Syrom78z3EQ5SbbB4sF7ey80etKII864WF64B81uRpH5t9jQTxeEu0ImbzRMqzVDZkVG9xD7nN1kuFw==
                    </P>
                    <Q>li7dzDacuo67Jg7mtqEm2TRuOMU=</Q>

<G>Z4Rxsnqc9E7pGknFFH2xqaryRPBaQ01khpMdLRQnG541Awtx/XPaF5Bpsy4pNWMOHCBiNU0NogpsQW5QvnlMpA==
                    </G>

<Y>w4C1+T2vRRGIsBuoD5DfRw76LdilsIs/5E4mdtb4MyWgFp/dS+ouFOQxa4dxhXEMBMdDdlI883Alr5DSiD/ybA==
                    </Y>
                  </DSAKeyValue>
                </KeyValue>
              </KeyInfo>
            </Signature>
          </saml2:Assertion>
        </oot:ObligationOfTrustStatement>
      </wsse:Security>
    </S:Header>
    <S:Body>
      <startNegotiation xmlns="http://provider.ws.xacml.xtmai.iris.salford.ac.uk/"/>
      <NegotiationState>
        <Service role="uk:ac:salford:iris:oot:service:role:client">
          <State value="uk:ac:salford:iris:oot:state:sao"/>
          <Status value="uk:ac:salford:iris:oot:status:ok"/>
        </Service>
      </NegotiationState>
    </S:Body>
  </S:Envelope>
```

E.2 .4 SAML attribute assertion sent to the service by the client described in test case 2 in chapter 9

```
   <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:oot="http://infosec.salford.ac.uk/names/oot"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope" xmlns:ws-
xacml="urn:oasis:names:tc:xacml:3.0:profile:webservices" xmlns:wsse="http://schema.xmlsoap.org/ws/2002/xx/sece"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <S:Header>
         <wsse:Security MustUnderstand="1">
            <oot:ObligationOfTrustStatement InResponseTo="1214903908126" IssuerInstant="2008-07-01T09:18:28Z"
ootID="1214903908534">
               <saml2:Assertion ID="1214903908535" IssueInstant="2008-07-01T09:18:28Z">
                  <saml2:Issuer>http://salford.ac.uk/client</saml2:Issuer>
                  <saml2:AttributeStatement>
                     <Attribute AttributeId="pcg-group" DataType="http://www.w3.org/2001/XMLSchema#string"
Issuer="http://salford.ac.uk/service" xmlns=""/>
                     <AttributeValue xmlns="">PCG Contractor</AttributeValue>
                  </saml2:AttributeStatement>
               </saml2:Assertion>
            </oot:ObligationOfTrustStatement>
         </wsse:Security>
      </S:Header>
      <S:Body>
      </ns2:startNegotiationResponse>
      <NegotiationState xmlns="">
         <Service role="uk:ac:salford:iris:oot:service:role:provider">
            <State value="uk:ac:salford:iris:oot:state:attribute"/>
            <Status value="uk:ac:salford:iris:oot:status:ok"/>
         </Service>
      </NegotiationState>
   </S:Body>
</S:Envelope>
```

E.2.5 Service XACML local policy used in test case 2.

```
 <?xml version="1.0" encoding="UTF-8"?>
<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      PolicyId="GeneratedPolicy"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.1:rule-combining-algorithm:ordered-permit-overrides">
 <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <AnyResource/>
    </Resources>
   <Actions>
    <AnyAction/>
   </Actions>
   </Target>

 <Rule RuleId="CommitRule" Effect="Permit">
<Target>
   <Subjects>
    <Subject>
     <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">tcb.sts.org</AttributeValue>
       <SubjectAttributeDesignator DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"
                        AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"/>
     </SubjectMatch>
     </Subject>
    <Subject>
     <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">xtmai.sts.co.uk</AttributeValue>
       <SubjectAttributeDesignator DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"
                        AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"/>
     </SubjectMatch>
    </Subject>
    <Subject>
     <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">pgr.salford.ac.uk</AttributeValue>
       <SubjectAttributeDesignator DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"
                        AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"/>
     </SubjectMatch>
    </Subject>
   </Subjects>
   <Resources>
    <Resource>
     <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">http://xtmai.tcbportal.com/</AttributeValue>
      <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#anyURI"
                        AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
     </ResourceMatch>
    </Resource>
   </Resources>
   <Actions>
     <Action>
      <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
       <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                          AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
      </ActionMatch>
     </Action>
    </Actions>
 </Target>
  <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
    <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                        AttributeId="pcg-groups"
                        Issuer="admin@pcg.org"/>
   </Apply>
   <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">PCG Contractor</AttributeValue>
  </Condition>
 </Rule>
 <Rule RuleId="FinalRule" Effect="Deny"/>
</Policy>
```

E. 3.1 XACML request context used in test case 3

```
  <Request>
    <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
        <AttributeValue>1214908470680@xtmai.sts.co.uk</AttributeValue>
      </Attribute>
    </Subject>
    <Resource>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
        <AttributeValue>http://xtmai.tcbportal.com/</AttributeValue>
      </Attribute>
    </Resource>
    <Action>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue>read</AttributeValue>
      </Attribute>
    </Action>
  </Request>
```

E. 3.2 NOB context sent to the client by the service, which is described in test case 3 of the chapter 9.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:oot="http://infosec.salford.ac.uk/names/oot"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope" xmlns:ws-
xacml="urn:oasis:names:tc:xacml:3.0:profile:webservices" xmlns:wsse="http://schema.xmlsoap.org/ws/2002/xx/sece"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <S:Header>
      <wsse:Security MustUnderstand="1">
        <oot:ObligationOfTrustQuery IssuerInstant="2008-07-01T09:45:12Z" ootID="1214905512917">
          <saml2:Assertion ID="1214905512932" IssueInstant="2008-07-01T09:45:12Z">
            <saml2:Issuer>http://xtmai.sts.salford.ac.uk/service</saml2:Issuer>
            <ws-xacml:XACMLPrivacyAssertion>
                <Requirements>
                  <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
                  <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
                     <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
                       <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
                             DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                     </Apply>
                   <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">30</AttributeValue>
                   </Apply>
                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                    <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                        AttributeId="pcg-groups"
                                        Issuer="admin@pcg.org"/>
                  </Apply>
                </Requirements>
                <Capabilities>
                   <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
                   <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
                      <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
                         <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
                              DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                      </Apply>
                      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">40</AttributeValue>
                   </Apply>
                </Capabilities>
            </ws-xacml:XACMLPrivacyAssertion>
          </saml2:Assertion>
        </oot:ObligationOfTrustQuery>
      </wsse:Security>
    </S:Header>
    <S:Body>
      <NegotiationState>
        <Service role="uk:ac:salford:iris:oot:service:role:client">
          <State value="uk:ac:salford:iris:oot:state:nob"/>
          <Status value="uk:ac:salford:iris:oot:status:ok"/>
        </Service>
      </NegotiationState>
    </S:Body>
</S:Envelope>
```

E. 3.3 NOB context sent to the service by the client, which is described in test case 3 of the chapter 9.

```
<S:Envelope xmlns S="http://schemas.xmlsoap.org/soap/envelope/" xmlns oot="http://infosec.salford.ac.uk/names/oot"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:soap="http://schemas xmlsoap.org/soap/envelope" xmlns:ws-
xacml="urn:oasis:names:tc:xacml:3 0:profile:webservices" xmlns.wsse="http.//schema.xmlsoap.org/ws/2002/xx/sece"
xmlns xsd="http://www w3.org/2001/XMLSchema" xmlns:xsi="http://www w3.org/2001/XMLSchema-instance">
    <S:Header>
      <wsse:Security MustUnderstand="1">
        <oot:ObligationOfTrustQuery InResponseTo="1214905512917" IssuerInstant="2008-07-01T09:45:13Z" ootID="1214905513059">
          <saml2:Assertion ID="1214905513059" IssueInstant="2008-07-01T09:45:13Z">
            <saml2.Issuer>http://xtmai.sts.salford.ac.uk/client</saml2 Issuer>
            <ws-xacml:XACMLPrivacyAssertion>
                  <Requirements>
                    <Vocabulary>urn:oasis.tc.xacml 3.0:vocabulary:xacml</Vocabulary>
                    <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
                       <Apply AttributeId="urn:oasis names:tc xacml:1.0:function:integer-less-than-or-equal">
                         <AttributeDesignator AttributeId="urn oasis:names:tc:xacml:3.0:action:max-data-rention-days"
                               DataType="http://www w3 org/2001/XMLSchema#integer"/>
                       </Apply>
                     <AttributeValue DataType="http.//www w3 org/2001/XMLSchema#integer">25</AttributeValue>
                     </Apply>
                  </Requirements>
                  <Capabilities>
                     <Vocabulary>urn:oasis tc:xacml 3.0:vocabulary:xacml</Vocabulary>
                     <Apply AttributeId="urn oasis names.tc xacml 3.0:function:must-be-present">
                        <Apply AttributeId="urn oasis names:tc xacml.1.0:function integer-less-than-or-equal">
                           <AttributeDesignator AttributeId="urn:oasis:names tc.xacml.3.0:action max-data-rention-days"
                                 DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                        </Apply>
                        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">25</AttributeValue>
                     </Apply>
                  <Apply FunctionId="urn oasis names.tc:xacml.1.0:function string-one-and-only">
                      <SubjectAttributeDesignator DataType="http://www.w3 org/2001/XMLSchema#string"
                                     AttributeId="pcg-groups"
                                     Issuer="admin@pcg.org"/>
                    </Apply>
                  </Capabilities>
            </ws-xacml:XACMLPrivacyAssertion>
          </saml2:Assertion>
        </oot.ObligationOfTrustQuery>
      </wsse Security>
    </S Header>
    <S:Body>
      <NegotiationState xmlns="">
        <Service role="uk:ac.salford.iris oot:service role:provider">
          <State value="uk:ac salford:iris:oot state.nob"/>
          <Status value="uk ac salford iris oot.status ok"/>
        </Service>
      </NegotiationState>
    </S:Body>
  </S:Envelope>
```

D. 3.4 SAO context sent to the service by the client, which is described in test case 3 in chapter

```
<S:Envelope xmlns:S="http://schemas xmlsoap.org/soap/envelope/" xmlns:oot="http://infosec.salford.ac uk/names/oot"
xmlns:saml2="urn:oasis:names tc.SAML:2.0:assertion" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope" xmlns:ws-
xacml="urn:oasis:names:tc:xacml:3.0:profile:webservices" xmlns:wsse="http://schema.xmlsoap.org/ws/2002/xx/sece"
xmlns:xsd="http://www w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <S:Header>
      <wsse:Security MustUnderstand="1">
        <oot:ObligationOfTrustStatement InResponseTo="1214905512917" IssuerInstant="2008-07-01T10.46 30Z" ootID="1214909190494">
          <saml2 Assertion ID="1214905840995" IssueInstant="2008-07-01T10:46:30Z">
            <saml2.Issuer>http://xtmai.sts.salford ac uk/client</saml2:Issuer>
            <ws-xacml XACMLPrivacyAssertion>
                  <Requirements>
                    <Vocabulary>urn:oasis:tc:xacml 3.0.vocabulary:xacml</Vocabulary>
                    <Apply AttributeId="urn.oasis names:tc:xacml 3.0:function:must-be-present">
                        <Apply AttributeId="urn oasis:names:tc:xacml.1.0:function:integer-less-than-or-equal">
                          <AttributeDesignator AttributeId="urn:oasis:names.tc:xacml:3.0:action:max-data-rention-days"
                  DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                        </Apply>
                     <AttributeValue DataType="http //www w3 org/2001/XMLSchema#integer">25</AttributeValue>
                     </Apply>
                  </Requirements>
                  <Capabilities>
                     <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary:xacml</Vocabulary>
                     <Apply AttributeId="urn:oasis:names:tc:xacml:3 0:function.must-be-present">
                        <Apply AttributeId="urn oasis names:tc:xacml:1.0:function:integer-less-than-or-equal">
                           <AttributeDesignator AttributeId="urn oasis:names:tc:xacml 3.0:action:max-data-rention-days"
                  DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                        </Apply>
                        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">25</AttributeValue>
                     </Apply>
                  <Apply FunctionId="urn:oasis:names tc:xacml:1.0:function:string-one-and-only">
                        <SubjectAttributeDesignator DataType="http://www w3 org/2001/XMLSchema#string"
                                    AttributeId="pcg-groups"
                                    Issuer="admin@pcg.org"/>
                    </Apply>
                  </Capabilities>
            </ws-xacml XACMLPrivacyAssertion>
            <Signature xmlns="http://www w3.org/2000/09/xmldsig#">
              <SignedInfo>
                <CanonicalizationMethod Algorithm="http //www.w3 org/2001/10/xml-exc-c14n#WithComments"/>
                <SignatureMethod Algorithm="http.//www w3 org/2000/09/xmldsig#dsa-sha1"/>
                <Reference URI="">
                  <Transforms>
                    <Transform Algorithm="http //www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                  </Transforms>
                  <DigestMethod Algorithm="http //www.w3.org/2000/09/xmldsig#sha1"/>
                  <DigestValue>Ld5Q8zmByMntGqJU0uSBs2s+asl=</DigestValue>
                </Reference>
              </SignedInfo>
              <SignatureValue>To/zjHbUx/E5HhfiUDJvP0DX1u182IYa+obVf4sCitN8r4emzthGCQ==</SignatureValue>
              <KeyInfo>
                <KeyValue>
                  <DSAKeyValue>
<P>/KaCzo4Syrom78z3EQ5SbbB4sF7ey80etKII864WF64B81uRpH5t9jQTxeEu0ImbzRMqzVDZkVG9xD7nN1kuFw== </P>
                    <Q>li7dzDacuo67Jg7mtqEm2TRuOMU=</Q>
                          <G>Z4Rxsnqc9E7pGknFFH2xqaryRPBaQ01khpMdLRQnG541Awtx/XPaF5Bpsy4pNWMOHCBiNU0NogpsQW5Qvnl
                          MpA==   </G>
                          <Y>TPu3ZNktIhCdrkaYCsUPIMX80Wr6uuqt0BZ4lelObcHwNrA4Ow0I0ka1w9StcLv8D2Zi/fh2GfWbubxfrDtH2Q==
                          </Y>
                  </DSAKeyValue>
                </KeyValue>
              </KeyInfo>
            </Signature>
          </saml2 Assertion>
        </oot:ObligationOfTrustStatement>
      </wsse Security>
    </S:Header>
    <S:Body>
      <NegotiationState xmlns="">
        <Service role="uk.ac.salford:iris.oot:service:role.provider">
          <State value="uk:ac:salford:iris:oot:state:sao"/>
          <Status value="uk:ac:salford:iris:oot:status:ok"/>
        </Service>
      </NegotiationState>
    </S:Body>
</S:Envelope>
```

D. 3.5 SAO context sent to the client by the service, which is described in test case 3 in chapter 9.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:oot="http://infosec.salford.ac.uk/names/oot"
xmlns:saml2="urn:oasis:names.tc:SAML:2.0:assertion" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope" xmlns:ws-
xacml="urn:oasis.names:tc:xacml:3.0:profile:webservices" xmlns:wsse="http://schema.xmlsoap.org/ws/2002/xx/sece"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <S:Header>
      <wsse:Security MustUnderstand="1">
        <oot:ObligationOfTrustStatement InResponseTo="1214909190494" IssuerInstant="2008-07-01T09:50:41Z" ootID="1214905840995">
          <saml2:Assertion ID="1214905841008" IssueInstant="2008-07-01T09:50:41Z">
            <saml2:Issuer>http://xtmai.sts.salford.ac.uk/service</saml2:Issuer>
            <ws-xacml:XACMLPrivacyAssertion>
                 <Requirements>
                    <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary.xacml</Vocabulary>
                    <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
                        <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
                          <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
                                 DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                        </Apply>
                     <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">30</AttributeValue>
                     </Apply>
                  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                       <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                      AttributeId="pcg-groups"
                                      Issuer="admin@pcg.org"/>
                    </Apply>
                  </Requirements>
                  <Capabilities>
                     <Vocabulary>urn:oasis:tc:xacml:3.0:vocabulary.xacml</Vocabulary>
                     <Apply AttributeId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
                        <Apply AttributeId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal">
                            <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:3.0:action:max-data-rention-days"
                                  DataType="http://www.w3.org/2001/XMLSchema#integer"/>
                        </Apply>
                        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">20</AttributeValue>
                     </Apply>
                  </Capabilities>
            </ws-xacml:XACMLPrivacyAssertion>
            <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
              <SignedInfo>
                <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#WithComments"/>
                <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
                <Reference URI="">
                  <Transforms>
                    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                  </Transforms>
                  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <DigestValue>+kwMrT/+5fab393BAMNrBtfR+Ss=</DigestValue>
               </Reference>
              </SignedInfo>
 <SignatureValue>icAGNNLacPcswP1TZQ2xvB0jtWhhvIdsSrA9C1+BCLACUlZSBgwN6Q==</SignatureValue>
              <KeyInfo>
                <KeyValue>
                  <DSAKeyValue>
                    <P>/KaCzo4Syrom78z3EQ5SbbB4sF7ey80etKII864WF64B81uRpH5t9jQTxeEu0ImbzRMqzVDZkVG9xD7nN1kuFw==
</P>
                                <Q>li7dzDacuo67Jg7mtqEm2TRuOMU=</Q>
                                <G>Z4Rxsnqc9E7pGknFFH2xqaryRPBaQ01khpMdLRQnG541Awtx/XPaF5Bpsy4pNWMOHCBiNU0NogpsQ
                                W5QvnlMpA== </G>
                                <Y>XGdQOI5Tp0IjKv/cgim+vupSrm0OLVLYxGixJqx7JdGPXFUM6+NShN8kb7TkxN5nzo6LW/iGBXXWsQ
                                jn7x02TQ== </Y>
                  </DSAKeyValue>
                </KeyValue>
              </KeyInfo>
            </Signature>
          </saml2:Assertion>
        </oot:ObligationOfTrustStatement>
      </wsse:Security>
    </S:Header>
    <S:Body>
<NegotiationState>
        <Service role="uk:ac:salford:iris:oot:service:role:client">
          <State value="uk:ac:salford:iris:oot:state.sao"/>
          <Status value="uk:ac:salford:iris:oot:status.ok"/>
        </Service>
      </NegotiationState>
    </S:Body>
  </S:Envelope>
```

D. 3.6 SAML attribute assertion sent to the service by the client, which is described in test case 3 in chapter 9

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:oot="http://infosec salford.ac.uk/names/oot"
xmlns:saml2="urn oasis:names tc SAML 2.0:assertion" xmlns soap="http://schemas.xmlsoap org/soap/envelope" xmlns:ws-
xacml="urn:oasis names:tc:xacml:3.0.profile:webservices" xmlns:wsse="http //schema xmlsoap.org/ws/2002/xx/sece"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http.//www.w3.org/2001/XMLSchema-instance">
    <S.Header>
      <wsse:Security MustUnderstand="1">
        <oot:ObligationOfTrustStatement InResponseTo="1214905840995" IssuerInstant="2008-07-01T10 50.17Z" ootID="1214909417608">
          <saml2.Assertion ID="1214909417608" IssueInstant="2008-07-01T10.50.17Z">
            <saml2:Issuer>http //salford.ac.uk/client</saml2.Issuer>
            <saml2:AttributeStatement>
              <Attribute AttributeId="group" DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http //salford.ac.uk/service"
xmlns=""/>
              <AttributeValue xmlns="">PCG Contractor
              </AttributeValue>
            </saml2:AttributeStatement>
          </saml2:Assertion>
        </oot:ObligationOfTrustStatement>
      </wsse Security>
    </S Header>
    <S Body>
      <NegotiationState xmlns="">
        <Service role="uk ac salford.iris.oot:service:role:provider">
          <State value="uk:ac salford iris:oot state attribute"/>
          <Status value="uk ac:salford iris:oot:status:ok"/>
        </Service>
      </NegotiationState>
    </S:Body>
  </S:Envelope>
```