Access Control Service Oriented Architecture Security

Yoon Jae Kim, yj1dreamer AT gmail.com (A project report written under the guidance of Prof. Raj Jain)

Download

Abstract

Service Oriented Architecture (SOA) is one of the most popular concepts to implement computing systems. However it faces many challenges to security and many standards and frameworks come out to support it. We focus especially on the access control system using SOA and represent what are the SAML and XACML and how they are applied for Portal and Web Services.

Keywords

Service Oriented Architecture, SOA, SOA Security, Web Service, Web Service Security, SAML, Security Assertion Markup Language, XACML, eXtensible Access Control Markup Language, access control

Table of Contents

1. Introduction to Services Oriented Architecture Security

One of the most popular IT trends is Service Oriented Architecture (SOA), which is defined as follows:

Service Oriented Architecture (SOA) is a design pattern which is composed of loosely coupled, discoverable, reusable, inter-operable platform agnostic services in which each of these services follow a well defined standard. Each of these services can be bound or unbound at any time and as needed. [Jamil08]

However, as defined, SOA has a loosely-coupled feature, which makes SOA open to the challenges of security. It means that SOA must meet several requirements. The main requirements are as follows[Candolin07]: service discovery, service authentication, user authentication, access control, confidentiality, integrity, availability, and privacy. To ensure security in a loosely-coupled SOA environment, the open standards communities that created Web services developed a number of security standards for Web services which is one of the most active and widely adopted implementation of SOA. Figure 1 depicts a notional reference model for Web services security standards. This reference model maps the different standards to the different functional layers of a typical Web service implementation.

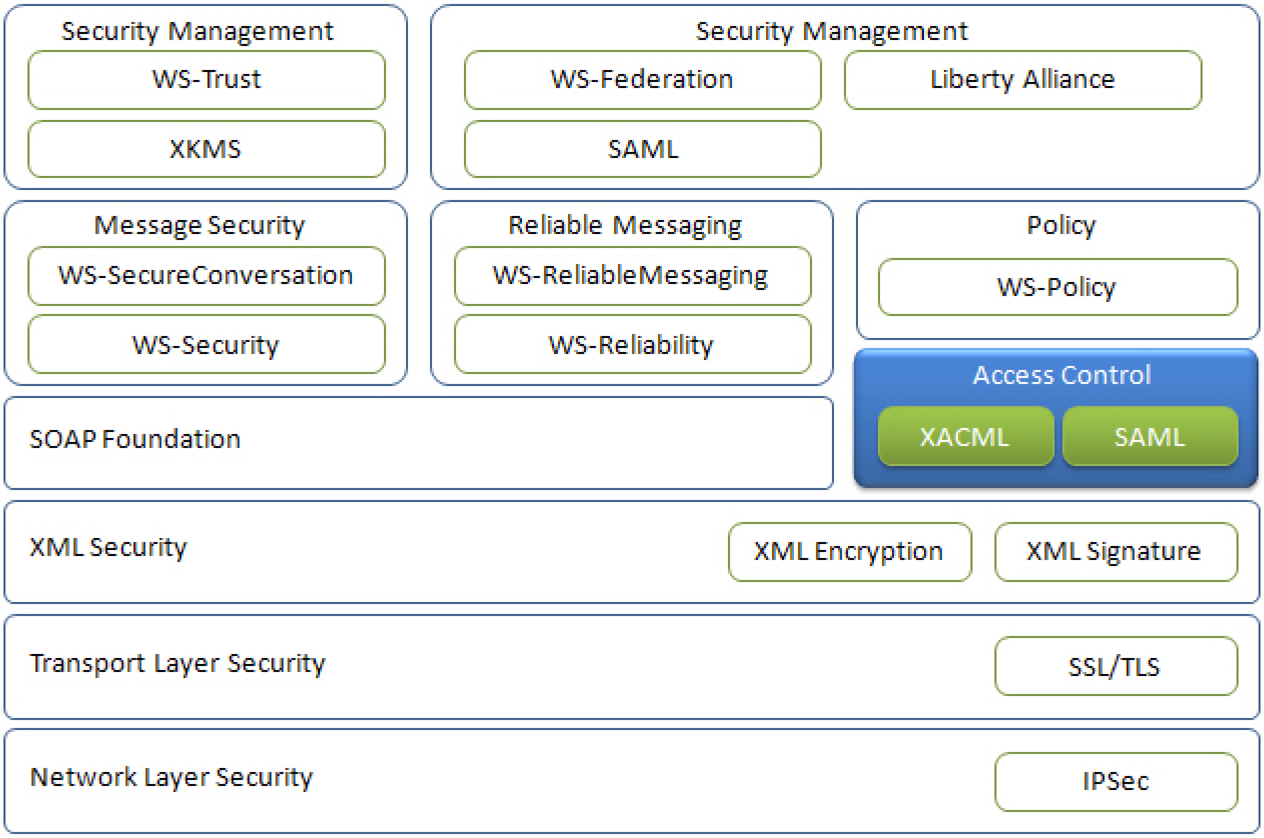

Figure 1. The Web Services Security Stack[Singhal07]

As described above, in the Web Services Security Stack the Security Assertion Markup Language (SAML) and the eXtensible Access Control Markup Language (XACML) are the standard for access control which means that when the service is requested by a user the service must enforce the specified security policy related to access control. We focus on access control in the Web Services security and represent what SAML and XACML are, how they work and where they are able to be applied together.

---

2. SAML (Security Assertion Markup Language)

SAML is an XML standard for exchanging authentication and authorization data between security domains. SAML has the feature like platform independent and is mainly applied to Single Sign-On (SSO).

2.1 What Is SAML?[Madsen05]

As many web sites are created and a lot of application systems are developed, federation is the prominent movement in identity management. Federation is defined as the establishment of business agreements, cryptographic trust, and user identifiers across security and policy domains to provide seamless cross-domain business interactions. As Web service based on XML turns up and provides integration between business entities by loose coupling at the application and messaging layer, federation can do so without the relation to the other's authentication and authorization infrastructure. To make this loose coupling possible at the identity management layer the standardized mechanisms and formats for exchanging security information is necessary and that is SAML.

SAML, created by the Security Services Technical Committee of the Organization for the Advancement of Structured Information Standards (OASIS), is a *an XML-based framework for communicating user authentication, entitlement, and attribute information. As its name suggests, SAML allows business entities to make assertions regarding the identity, attributes, and entitlements of a subject (an entity that is often a human user) to other entities, such as a partner company or another enterprise application.* [Madsen05] SAML is a flexible and extensible protocol designed to be used - and customized if necessary - by other standards.

2.2 SAML Architecture[Ragouzis08]

SAML consists of six components as follows: assertions, protocols, bindings, profiles, metadata, authentication context. The relationship between these components is similar to building-blocks and when they are put together they allow a number of use cases to be supported such as web single sign-on use case and identity federation use case. The components mainly enable to transfer secure information like identity, authentication, and authorization information between trusted entities.

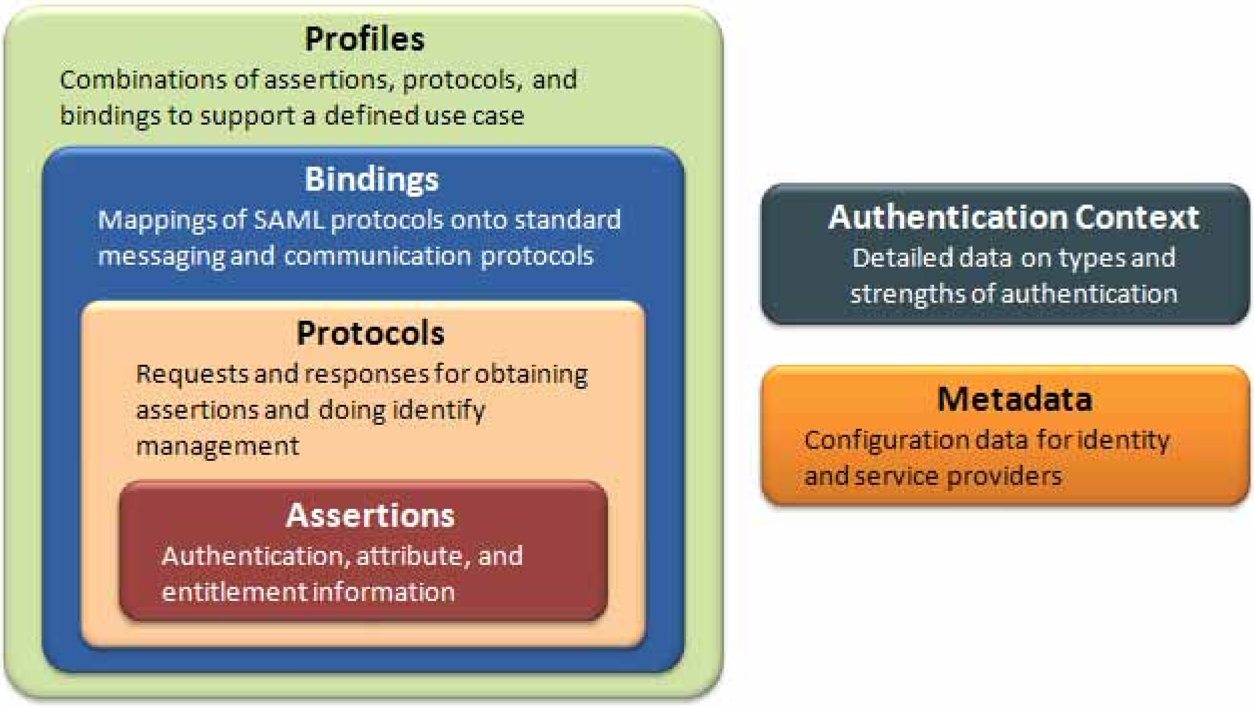
Profiles
Combinations of assertions, protocols, and bindings to support a defined use case
Bindings
Mappings of SAML protocols onto standard messaging and communication protocols
Protocols
Requests and responses for obtaining assertions and doing identify management
Assertions
Authentication, attribute, and entitlement information
Authentication Context
Detailed data on types and strengths of authentication
Metadata
Configuration data for identity and service providers

Figure 2. the relationship between basic SAML Concepts

- SAML assertions contain identifying information made by a SAML authority. In SAML, there are three assertions: authentication, attribute, and authorization. Authentication assertion validates that the specified subject is authenticated by a particular means at a particular time and is made by a SAML authority called an identity provider. Attribute assertion contains specific information about the specified subject. And authorization assertion identifies what the specified subject is authorized to do

- SAML protocols define how SAML asks for and receives assertions and the structure and contents of SAML protocols are defined by the SAML-defined protocol XML schema.
- SAML bindings define how SAML request-response message exchanges are mapped to communication protocols like Simple Object Access Protocol (SOAP). SAML works with multiple protocols including Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP) and so on.
- SAML profiles define constraints and/or extensions to satisfy the specific use case of SAML. For example, the Web SSO Profile details how SAML authentication assertions are exchanged between entities and what the constraints of SAML protocols and binding are. Another type of SAML profile is an attribute profile which establishes specific rules for interpretation of attributes in SAML attribute assertions. For instance, X.500/LDAP profile details how to carry X.500/LDAP attributes within SAML attribute assertions.
- SAML metadata defines a way to express and share configuration information between SAML entities. For instance, an entity's supported SAML bindings, operational roles (IDP, SP, etc), identifier information, supporting identity attributes, and key information for encryption and signing can be expressed using SAML metadata XML documents. SAML Metadata is defined by its own XML schema.
- In a number of situations, a service provider may need to have detailed information regarding the type and strength of authentication that a user employed when they authenticated at an identity provider. A SAML authentication context is used in (or referred to from) an assertion's authentication statement to carry this information. An SP can also include an authentication context in a request to an IdP to request that the user be authenticated using a specific set of authentication requirements, such as a multi-factor authentication.

This SAML architecture, which is composed of building-block components, provides the flexibility and the extensibility to implement the system and makes it possible to support various business use cases.

2.3 The Advantages of SAML

The previous part represents that SAML has the build-blocks architecture, which gives SAML the benefits as follows:

- Platform neutrality - SAML separates the security framework from platform architecture and specific vendor implementation. Making security more independent of application logic is an important tenet of Service-Oriented Architecture.
- Loose coupling of directories - SAML does not require user information to be maintained and synchronized between directories.
- Improved online experience for end users - SAML enables single sign-on by allowing users to authenticate at an identity provider and then access service providers without additional authentication. In addition, identity federation (linking of multiple identities) with SAML allows for a better-customized user experience at each service while promoting privacy.
- Reduced administrative costs for service providers - Using SAML to "reuse" a single act of authentication (such as logging in with a username and password) multiple times across multiple services can reduce the cost of maintaining account information. This burden is transferred to the identity provider.

2.4 The Usages of SAML

As befits a general framework for communicating security and identity information, SAML is being applied in a number of different ways, the major ones of which are presented here.

Web Single Sign-On
In web SSO, a user authenticates to one web site and then, without additional authentication, is able to access some personalized or customized resources at another site. SAML enables web SSO through the communication of an authentication assertion from the first site to the second which, if confident of the origin of the assertion, can choose to log in the user as if they had authenticated directly. A principal authenticates at the identity provider and is subsequently appropriately recognized (and given corresponding access/service) at the service provider.[Google]

For example, Google made SAML Single Sign-On (SSO) Service for Google Apps. And Google Apps provides a SAML-based Single Sign-On (SSO) service that offers partner companies with full control over the authorization and authentication of hosted user accounts that can access web-based applications like Gmail or Google Calendar. As the service provider Google offers services as Gmail and Start Pages and partner companies control account information as identity provider.

Attribute-Based Authorization
Similar to the Web SSO scenario, the attribute-based authorization model has one web site communicating identity information about a subject to another web site in support of some transaction.

However, the identity information may be some characteristic of the subject (such as a person's role in a B2B scenario) rather than, or in addition to, information about when and how the person was authenticated. The attribute-based authorization model is important when the individual's particular identity is either not important, should not be shared for privacy reasons, or is insufficient on its own.

Securing Web Services
SAML assertions can be used within SOAP messages in order to convey security and identity information between actors in web service interactions. The SAML Token Profile produced by the OASIS Web Services Security (WSS) TC specifies how SAML assertions should be used for this purpose with the WS-Security framework. The Liberty Alliance's Identity Web Service Framework (ID-WSF) builds on these specifications to use SAML assertions for enabling secure and privacy-respecting access to web services.

WS-Trust, one component of the private WS-* framework initiative, proposes protocols for the exchange and validation of security tokens used as described within WS-Security. SAML assertions are one such supported security token format.

---

3. XACML (eXtensible Access Control Markup Language)

XACML is a declarative access control policy language implemented in XML as well as a processing model that describes how to interpret the policies. Compared with an access control list (ACL), XACML provides flexible, generic and extensible access control.

3.1 What is XACML?

XACML is an OASIS standard that describes both a policy language implemented in XML and an access control decision request/response language implemented in XML. The policy language details general access control requirements, and has standard extension points for defining new functions, data types, combining logic, etc. The request/response language lets you form a query to ask whether or not a given action should be allowed, and interpret the result. The response always includes an answer about whether the request should be allowed using one of four values: Permit, Deny, Indeterminate (an error occurred or some required value was missing, so a decision cannot be made) or Not Applicable (the request can't be answered by this service).

3.2 How does XACML work?

The XACML profile specifies five main actors to handle access decisions: Policy Enforcement Point (PEP), Policy Administration Point (PAP), Policy Decision Point (PDP), Policy Information Point (PIP), and a context handler.[Moses05]

- Policy Administration Point (PAP): PAP is the repository for the policies and provides the policies to the Policy Decision Point (PDP).
- Policy Enforcement Point (PEP): PEP is actually the interface of the whole environment to the outside world. It receives the access requests and evaluates them with the help of the other actors and permits or denies the access to the resource.
- Policy Decision Point (PDP): PDP is the main decision point for the access requests. It collects all the necessary information from other actors and concludes a decision.
- Policy Information Point (PIP): PIP is the point where the necessary attributes for the policy evaluation are retrieved from several external or internal actors. The attributes can be retrieved from the resource to be accessed, environment (e.g., time), subjects, and so forth.

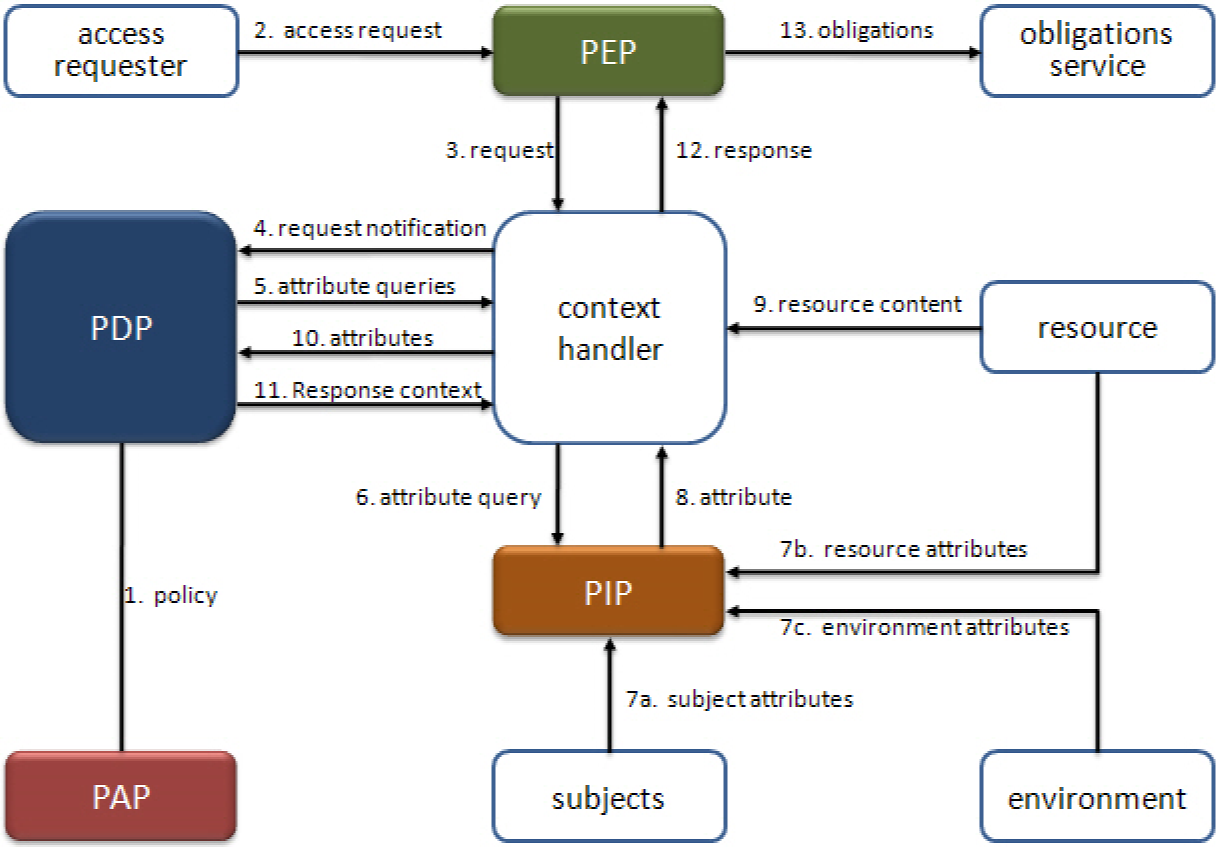

Figure 3. XACML Actors and Data flow

Figure 3 illustrates these actors and information flow. As can be seen in the figure, the PAP writes Polices and PolicySets and makes them available to the PDP. These Policies or PolicySets shows the complete policy for a particular target. The PEP is the component where the request is received when access requester wants to take some action on a resource and make the request. In this part, the attributes in the request may be in the format of the application environment (e.g., SAML, etc.). The PEP sends the request to the Context Handler.

Context Handler maps the request and attributes to the XACML Request context and sends the request to the PDP. While evaluating the request, the PDP needs some attributes and sends the attribute queries to the Context Handler. The Context Handler collects these attributes by the help of the PIP from the resources, subjects, and the environment. After evaluation, the PDP sends the XACML Response to the Context Handler and the Context Handler translates the Response context to the native response format of the application environment and sends it to PEP. The PEP fulfills the obligations if they exist and applies the authorization decision that PDP concludes.[Periorellis07]

3.3 XACML context

XACML is intended to be suitable for a variety of application environments. The core language is insulated from the application environment by the XACML context, as shown in Figure 4, in which the scope of the XACML specification is indicated by the shaded area. Therefore, applications can use other representations like SAML, which is the most suitable one for the attributes. Then the PEP applications convert these attribute representations to the XACML context attributes. The XACML context is defined in XML schema, describing a canonical representation for the inputs and outputs of the PDP.

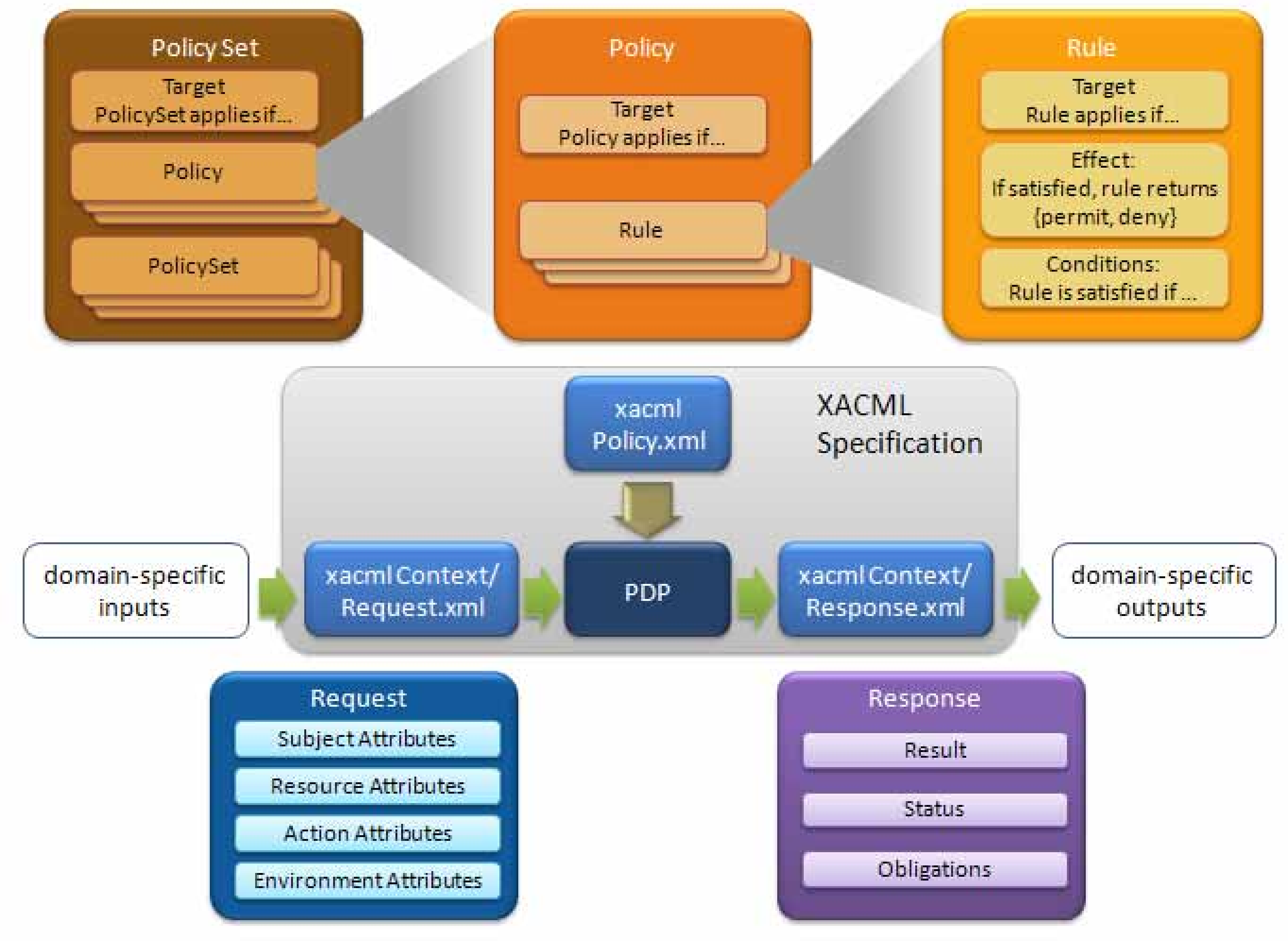

Figure 4. XACML context

A Request element contains four components as Subject, Resource, Action, and Environment. One request element has only one collection of resource and action attributes, and at most one collection of environment attributes. But there may be multiple collections of subject attributes. Subject attribute contains subject's details such as name, e-mail, role and so on. Resource attribute details the resource for which access is requested and action attribute specifies the requested action to be performed on resource such as read or wire. Also, Environment attribute is optional and contains attributes of environment.

A Response element represents the authorization decision information made by PDP. It contains one or more Result attributes. Each result includes a Decision such as Permit, Deny, NotApplicable, or Indeterminate, some Status information which gives the errors occurred and their descriptions while evaluating the request and optionally one or more Obligations which specifies tasks in the PolicySet and Policy elements in the policy description which should be performed before granting or denying access.

A Rule element defines the target elements to which the rule is applied and details conditions to apply the rule and has three components such as target, effect, and condition. A target element specifies the resources, subjects, actions and the environment to which the rule is applied. A condition element shows the conditions to apply the rule and a effect is the consequence of the rule as either permit or deny.

A policy is the set of rules which are combined with some algorithms. These algorithms are called Rule-combining algorithms. For instance "Permit Override" algorithm allows the policy to evaluate to "Permit" if any rule in the policy evaluates to "Permit". A policy also contains target elements which shows the subjects, resources, actions, environment that policy is applied.

A PolicySet consists of Policies and PolicySets combined with policy-combined algorithm. It has also target like a Policy.

The XACML context shows how flexible and suitable the XACML is for various application. This feature makes it possible that XACML is applied to access control system with SAML. Section 4 shows the more detailed.

3.4 The Advantage of XACML

There are many existing proprietary and application-specific languages for doing this kind of thing but XACML has several points in its favor[Sun]:

- It's standard. By using a standard language, you're using something that has been reviewed by a large community of experts and users, you don't need to roll your own system each time, and you don't need to think about all the tricky issues involved in designing a new language. Plus, as XACML becomes more widely deployed, it will be easier to interoperate with other applications using the same standard language.
- It's generic. This means that rather than trying to provide access control for a particular environment or a specific kind of resource, it can be used in any environment. One policy can be written which can then be used by many different kinds of applications, and when one common language is used, policy management becomes much easier.
- It's distributed. This means that a policy can be written which in turn refers to other policies kept in arbitrary locations. The result is that rather than having to manage a single monolithic policy, different people or groups can manage separate sub-policies as appropriate, and XACML knows how to correctly combine the results from these different policies into one decision.
- It's powerful. While there are many ways the base language can be extended, many environments will not need to do so. The standard language already supports a wide variety of data types, functions, and rules about combining the results of different policies. In addition to this, there are already standards groups working on extensions and profiles that will hook XACML into other standards like SAML and LDAP, which will increase the number of ways that XACML can be used.

---

4. Access Control using SAML and XACML

In an access control model for Web Service SAML makes a role to protect, transport, and request XACML instances and XACML defines the information exchanging between each entity such as request, response and policy. This model can be applied to access control between Portal and Web Services.

4.1 SAML 2.0 Profile of XACML 2.0

XACML is a powerful standard language that defines schemas for authorization policies and for authorization decision requests/responses and that specifies how to evaluate policies, but confines its scope to the language elements used directly by the PDP and does not specify protocols or transport mechanisms. XACML also does not define how to implement PEP, PAP, AA, context handler, or Repository. For example, the Policy Enforcement Point (PEP) sends the request to the Policy Decision Point (PDP) with the "Request" element defined in the XACML context. But XACML files can serve as a standard format for exchanging information between these entities when combined with other standards.

SAML is one standard suitable for providing the assertion and protocol mechanisms and specifies schemas for carrying the security and authorization related information and have the bindings to basic transportation mechanisms. Therefore, OASIS publishes a SAML profile for the XACML (OASIS, 2005)[Anderson05] to carry the XACML messages between the XACML actors. This profile defines the usage of SAML 2.0 to protect, store, transport, request and respond with XACML instances and other information. It contains largely four categories.

First, this profile specifies how to use SAML Attributes in an XACML system. This category contains three standard SAML elements such as SAML Attribute, SAML AttributeStatement and SAML Assertion, two standard SAML protocol such as SAML AttributeQuery and SAML Response, and one new SAML extension element, XACMLAssertion. In an XACML system, SAML Attribute may be used to store and to transmit attribute values and must be transformed into an XACML Attribute before used in an XACML Request Context. Also SAML AttributeStatement may be used to hold SAML Attribute instances. A SAML Assertion may be used to hold SAML AttributeStatement instances in an XACML system, either in an Attribute Repository or in a SAML Response to a SAML AttributeQuery. To transform a SAML Attribute into an XACML Attribute the SAML Assertion includes information that is required and a SAML Assertion or an XACMLAssertion instance contains a SAML Attribute. An XACMLAssertion is an alternative to the SAML Assertion and allows inclusion of XACML Statement instances and inclusion of other XACMLAssertion instance as advice. An XACML PDP or PEP use SAML AttributeQuery to request SAML Attribute instances from an Attribute Authority for use in an XACML Request Context and in response to it SAML Response shall be used to return SAML Attribute instances.

Second, this profile represent the use of SAML for use in requesting, responding with, storing, and transmitting authorization decisions in an XACML system. This category contains XACMLAuthzDecisionStatement, XACMLAssertion, XACMLAuthzDecisionQuery, and XACMLResponse. In this profile, XACMLAuthzDecisionStatement and XACMLAssertion are new SAML extension elements and the others are new SAML extension protocol elements. In an XACML system, XACMLAuthzDecisionSatement may be used to contain XACML authorization decisions for storage or transmission and XACMLAssertion may be used to contain XACMLAuthzDecisionStatement instances for storage or transmission. Also a PEP may use XACMLAuthzDecisionQuery to request an authorization decision from an XACML PDP and an XACML PDP may use XACMLResponse to return authorization decisions in response to an XACMLAuthzDecisionQuery.

Then, this profile shows the use of SAML for use in requesting, responding with, storing and transmitting XACML policies. This category includes four new SAML extensions; XACMLPolicyStatement, XACMLAssertion, XACMLPolicyQuery and XACMLResponse. In an XACML system, XACMLPolicyStatement may hold XACML policies for storage or transmission and XACMLAssertion may hold XACMLPolicySatement instances for storage or transmission. And a PDP or other application uses XACMLPolicyQuery to request XACML from a PAP. Also PAP uses XACMLResponse to return policies in response to an XACMLPolicyQuery.

Finally, this profile details the use of XACMLAssertion instances as advice in other Assertion. This category consists of XACML Advice, which is a new SAML extension element in this profile that may be used for including XACMLAssertion instances as advice in another XACMLAssertion, and XACMLAssertion which is a new SAML extension element that may be used to hold on XACMLAdvice instance along with SAML Statement or XACML extension Statement instance.

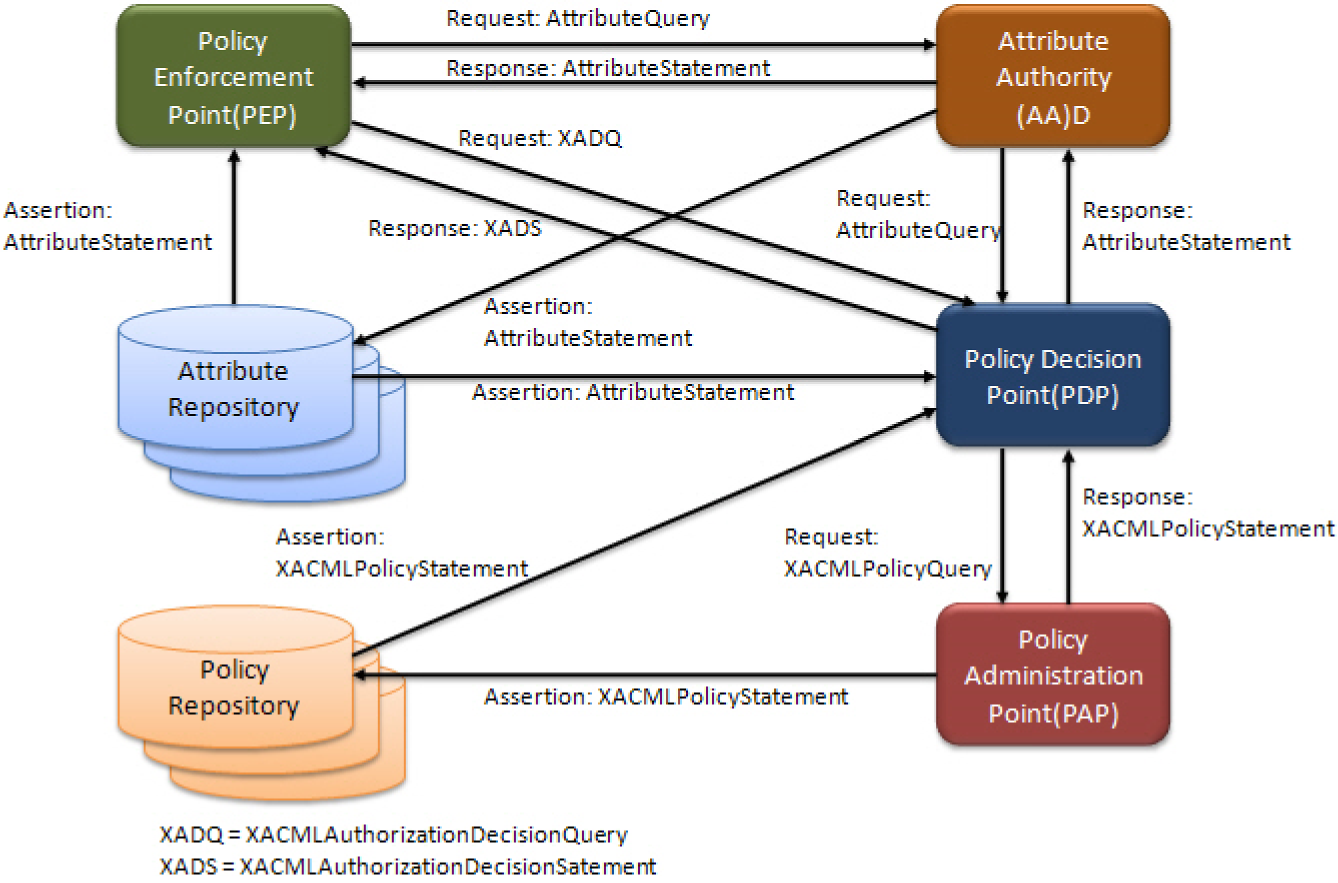
Policy Enforcement Point(PEP)
Request: AttributeQuery
Response: AttributeStatement
Attribute Authority (AA)D
Request: XADQ
Response: XADS
Assertion: AttributeStatement
Request: AttributeQuery
Response: AttributeStatement
Attribute Repository
Assertion: AttributeStatement
Assertion: AttributeStatement
Policy Decision Point(PDP)
Assertion: XACMLPolicyStatement
Request: XACMLPolicyQuery
Response: XACMLPolicyStatement
Policy Repository
Assertion: XACMLPolicyStatement
Policy Administration Point(PAP)
XADQ = XACMLAuthorizationDecisionQuery
XADS = XACMLAuthorizationDecisionSatement

Figure 5. Components and messages in as integration of SAML with an XACML system

Figure 5 describes the XACML use model and the messages that can be used to communicate between the various components. Statements are carried in SAML or XACML Assertions, and Assertions are carried in SAML or XACML Responses. Not all components or messages will be used in every implementation. Next subsection shows the practical example of this model.

4.2 SAML/XACML based Access Control between Portal and Web Services

Many web-based portals aggregate many various types of information by interacting with remote Web services. So access control between portal and Web Service is very important issue. In that case A SAML/XACML based Access Control is a very implementable solution of many possible solutions which are widely exposed[YIN07]. In order to transparently transfer the identity information between Portal and Web service, Web service handlers are adopted on both sides. On the Portal side, handlers are used to create a SAML authentication assertion containing the user identity, attach the assertion as a XML token of WS-Security in the SOAP header with the digital signature of the trusty Portal. On the Web service side, handlers act as PEP to intercept the received message and to verify the signature and translate the required information (subject, action and resource) to construct a XACML request which is forwarded to PDP. When PDP makes an access decision, PEP parses the returned XACML response and decides if the request for Web service is permitted or denied. The architecture of the proposed solution is shown in Figure 6.

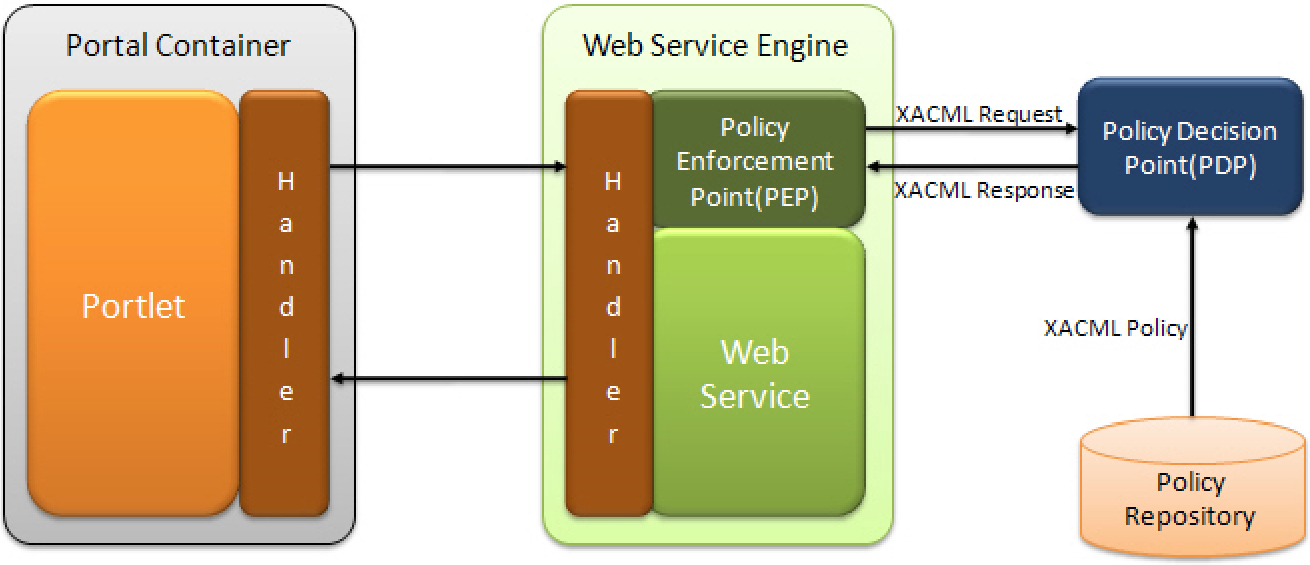

Figure 6. Architecture of the SAML/XACML based System

The steps of communication between Portal and Web services are described in detail as follows:

1. A user login the Portal and triggers the Portal to invoke the back-end Web service.
2. The handler intercepts the SOAP message and embeds a SAML authentication assertion in the SOAP header with the digital signature of the Portal. The assertion contains the user identity provided by Portal. Then the message is sent out. This step achieves to extend the authentication of the user from Portal to external Web service.
3. The handler on the Web service side acts as the SAML PEP. It intercepts the received SOAP message and validates the signature. Then it determines the assigned role of the requester as subject through identity management module, the operation name extracted from the SOAP body as action, and the name of Web service as resource. With these required factors, SAML PEP generates a XACMLAuthzDecisionQuery and sends it to SAML PDP.
4. SAML PDP is the point to actually make access decision. It retrieves the policies from policy storage, such as file system or a database, like dbXML, a native XML database to store XML, and then evaluates whether the request is granted. Finally it constructs a XACMLAuthzDecisionStatement as a SAML response containing a XACML response context and sends it back to SAML PEP in response to the XACMLAuthzDecisionQuery. This step achieves the authorization of the user on the Web service side.
5. SAML PEP parses the returned response to extract the decision value. If the result is PERMIT, the requester has the privilege to invoke Web service, or the process is terminated.
6. The result of the invocation to the Web service is returned to Portal.

This access control mechanism is very effective and flexible to change policies about the entity. In addition it is more suitable for dynamic and large-scale application domain.

5. Summary

SOA provides the solution to the system which consists of many tangled applications because they are loose-coupled. However, SOA faces the threats about security and need to meet several requirements such as service discovery, service authentication, user authentication, access control, service usage, confidentiality, integrity, availability, privacy and trust management. Then the open standards communities developed a number of security standards based on XML language in order to meet security issues.

Focusing on access control we represent SAML and XACML which are developed by OASIS. SAML is an XML-based framework for exchanging authentication and authorization data. Because SAML has much strength such as platform neutrality, loose coupling of directories, improved online experience for end user, reduced administrative costs for service providers and risk transference. Also SAML is being applied in Web Single Sign-On, Attribute-Based Authorization, and Securing Web Services.

XACML defines XML files which contains access control policy and access control decision request/ response. Policy Decision Point (PDP) looks at the request from Policy Enforcement Point (PEP) and finds some policy applying to the request from Policy Administration Point (PAP) and returns the response about whether access should be granted to PEP.

XACML defines the content of Request/Response messages but does not define protocols or transport mechanisms, which SAML provides by defining schemas for use in requesting and responding with various types of security assertions. This SAML/XACML based access control is a very powerful and practical solution for dynamic and large-scale application domain because it is easier to change and maintain policies. So it can extend the authentication and authorization mechanism within a portal to external Web services.

References

[Candolin07] Candolin, Catharina, "A Security Framework for Service Oriented Architectures", Military Communications Conference, 2007. MILCOM 2007. IEEE, 29-31 Oct. 2007, pp.1-6 http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4455332

[Singhal07] Singhal , Anoop, "Web Services Security: Challenges and Techniques" policy, Eighth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'07), 2007, pp.282 http://www2.computer.org/portal/web/csdl/doi/10.1109/POLICY.2007.50

[Madsen05] Madsen, Paul, et al., "SAML V2.0 Executive Overview", OASIS Committee Draft, 12 April 2005 http://www.oasis-open.org/committees/download.php/13525/sstc-saml-exec-overview-2.0-cd-01-2col.pdf

[Ragouzis08] Ragouzis, Nick, et al., "Security Assertion Markup Language (SAML) V2.0 Technical Overview", Committee Draft 02, 25, March 2008, http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf

[Sun] "Sun's XACML Implementation", July 2004, http://sunxacml.sourceforge.net/guide.html

[Moses05] Moses, Tim, et al., "eXtensible Access Control Markup Language(XACML) Version 2.0", OASIS Standard, 1 Feb 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[Periorellis07] Periorellis,Panos , "Securing Web Services: Practical Usage of Standards and Specifications", Idea Group Inc(IGI), 2007. http://books.google.com/books?id=zX2N7fWTJOUC

[YIN07] Yin, Hao, et al., "A SAML/XACML Based Access Control between Portal and Web Services", Data, Privacy, and E-Commerce, 2007. ISDPE 2007. The First International Symposium on, Nov. 2007, pp 356-360 http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4402710

[Anderson05] Anderson, Anne, et al., "SAML 2.0 profile of XACML v2.0", OASIS Standard, 1 Feb 2005 http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf

[Jamil08] Jamil, Ejaz, et al., "What really is SOA. A comparison with Cloud Computing, Web 2.0, SaaS, WOA, Web Services, PassS and others", SOALIB, 12 Dec 2008. http://soalib.com/docs/whitepaper/SoalibWhitePaper_SOAJargon.pdf

[Google] SAML Single Sign-On (SSO) Service for Google Apps, http://code.google.com/apis/apps/sso/saml_reference_implementation.html

---

List of Acronyms

AA	Attribute Authority
B2B	Business-to-business
CORBA	Common Object Request Broker Architecture
DCE	Distributed Computing Environment
GSA	General Services Administration
IDP	General Services Administration
J2SE	Java Platform Standard Edition
ID-WSF	Identity Web Services Framework
LDAP	Lightweight Directory Access Protocol
OASIS	the Organization for the Advancement of Structured Information Standards
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Point Information Point
SAML	Security Assertion Markup Language
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SP	Service Provider
SSL	Secure Sockets Layer
SSO	Single Sign-On
TLS	Transport Layer Security
WSS	Web Security Service
XACML	eXtensible Access Control Markup Language
XKMS	XML Key Management Specification
XML	eXtensible Markup Language
XSLT	eXtensible Markup Language

Last Modified: April, 19, 2009
This and other papers on latest advances in network security are available on line at http://www.cse.wustl.edu/~jain/cse571-09/index.html
SHARE Back to Raj Jain's Home Page