GWD-R (proposed)
Category – recommendation
GGF working group on
OGSA Authorization

Mary Thompson, Lawrence Berkeley Nat'l Lab
Von Welch, University of Chicago
Markus Lorch, Virgina Tech
Rebekah Lepro, NASA Ames
David Chadwick, Univeristy of Salford
Vincenzo Ciaschini, INFN CNAF

February 24, 2004

GGF DOCUMENT SUBMISSION CHECKLIST (include as front page of submission)	
	COMPLETED (X) - Date
1. Author name(s), institution(s), and contact information	(X) – 2004-02-24
2. Date (original and, where applicable, latest revision date)	(X) – 2004-02-24
3. Title, table of contents, clearly numbered sections	(X) – 2004-02-24
4. Security Considerations section	(X) – 2003-09-03
5. GGF Copyright statement inserted (See below)	(X) – 2003-09-03
6. GGF Intellectual Property statement inserted. (See below) **NOTE that authors should read the statement.**	(X) – 2003-09-03
7. Document format - The GGF document format to be used for both GWD's and GFD's is available in MSWord, RTF, and PDF formats. (note that font type is not part of the requirement, however authors should avoid font sizes smaller than 10pt).	(X) – 2004-02-24

Mary Thompson, Lawrence Berkeley Nat'l Lab
Von Welch, University of Chicago
Markus Lorch, Virgina Tech
Rebekah Lepro, NASA Ames
David Chadwick, Univeristy of Salford
Vincenzo Ciaschini, INFN CNAF

February 24, 2004

Discussion points for GGF10

Add VOMS FQAN as a type of attribute. /VO[/group[/subgroup(s)]][/Role=role][/Capability=cap]

I don't think we have ever resolved the namespace issue for attributes. SAML includes a namespace as part of an attribute name, XACML does not. Should we define a rule for concatenating namespace and name to get a single identifier?

Vincenzo added a HolderIdentification extension to the X.509 AC. This would allow an AA to add the information that the AA knows the holder of this AC by this additional (uniquely defined by the AA) identifier. This could be added into the SAML profile as an additional attribute could be bound to the official identity(s) of the person. It not clear that the PDP could know that the AC for a different official name applies to a requestor, as it knows the requestor only by another of its official (authenticated) name.

GWD-R (proposed)
Category – recommendation
GGF working group on
OGSA Authorization

Mary Thompson, Lawrence Berkeley Nat'l Lab
Von Welch, University of Chicago
Markus Lorch, Virgina Tech
Rebekah Lepro, NASA Ames
David Chadwick, Univeristy of Salford
Vincenzo Ciaschini, INFN CNAF

February 24, 2004

Attributes used in OGSA Authorization

Status of This Memo

This document has been submitted to the Global Grid Forum OGSA OGSA-Authz Working Group for consideration as recommendations document in that area of OGSA authorization. The latest version of this document can be found at: https://forge.gridforum.org/projects/ogsa-authz/document/

Abstract

This document specifies elements and vocabulary for expressing descriptive and privilege attributes to be used in the context of Open Grid Services Architecture (OGSA) authorization. The intention of defining standard formats and meanings (vocabulary) for these assertions is to facilitate compatibility between issuers of attribute assertions and the authorization systems that consume them. Profiles for specifying attribute assertions using SAML AttributeAssertions and X.509 attribute certificates are also included.

Contents

1. Introduction

This document is a companion to the “OGSA Authorization Requirements” GWD [OGSA-authz-req] and “Use of SAML for OGSA Authorization” [OGSA-service] and assumes that the reader is familiar with those papers. Many terms used in this document are defined in a common glossary that is included at the end of this document.

Most authorization systems that make decisions based on access control policy consider attributes of an initiator in addition to identity. Basing all access control on the initiator identity alone requires an extremely verbose and inflexible policy that does not scale well as more principals are added to the policy.

The intention of this document is to allow for interoperability between attribute authorities (AA) which issue attribute assertions, the policy writers who define access policy, and access decision functions (ADFs) that make decisions based on the initiator’s attributes and resource policy or access enforcement functions (AEF) in the case of privilege attributes. In a typical Grid environment there may be several authorities that assert attributes for users. Various domains will want to write authorization policy based on such attributes. Standard methods for discovering, guaranteeing integrity and transporting these assertions as well as common formats and vocabularies for expressing their assertion semantics are needed to enable the various pieces of a Grid to interact.

A number of methods for requesting and encoding attributes already exist (e.g., X.509 attribute certificates [RFC3281], SAML attribute assertions [SAML] and XACML attributes [XACML]. This document does not intend to define a new method or dictate the use of an existing method. Instead, it documents the functionalities needed to support OGSA authorization and defines profiles for encoding these functions using SAML attribute assertions and X.509 attribute certificates. It is expected that other profiles will be defined for the use of other mechanisms in OGSA.

Section 2 defines the conventions and namespaces used in this document. Section 3 presents an overview of the requirements for the use and content of attributes in the OGSA authorization context. Section 4 provides a non-normative discussion of current attribute mechanisms. Section 5 contains a normative set of definitions for attributes to be used in OGSA authorization. Section 6 contains a normative profile for expressing OGSA attribute assertions using SAML. Section 7 contains a normative profile for expressing the attribute assertions using X.509 attribute certificates

2. Conventions used in this Specification

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

The following namespace prefixes may be used in XML examples in this document. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 1: Name spaces used in this specification.

Prefix	Namespace
ogsa-saml	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/
operation	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/operation
sde-read	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/sde/read
sde-modify	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/sde/modify
wildcard	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/wildcard

saml	urn:oasis:names:tc:SAML:1.0:assertion
samlp	urn:oasis:names:tc:SAML:1.0:protocol

3. OGSA Use and Requirements for Attributes.

Attributes provide information about entities that can be used in addition to [or in lieu of] the entity's identity to make authorization decisions. There are two major classes of attributes: *descriptive attributes* and *privilege attributes*. Descriptive attributes associate a characteristic with an entity, while privilege attributes define an entity's rights with respect to a resource. Descriptive attributes are used in combination with access policy to yield rights to specific resources. Descriptive attributes can be characterized by the entity with which they are associated. For example, attributes associated with the initiator of an action are called subject attributes. Those that define attributes of resources are called resource attributes while those that describe attributes of the environment are called environment attributes. This document will focus on subject and privilege attributes. However, it will keep in mind the other types of descriptive attributes when deciding on the components of an attribute, since it is desirable to have a common format to represent arbitrary attributes.

Privilege attributes bind directly applicable access right(s) for one or more resources to a subject. Privilege attributes hold many similarities to an authorization decision. A privilege is an attribute that is tied to a tuple {subject, action, resource}. Privilege attributes that are presented along with a matching request to the target resource guarantee that access will be granted unless the privilege attributes are revoked or otherwise invalidated or superseded. Of course, authenticity and issuer authoritativeness have to be verified in the same way as needed for subject attributes.

Subject attributes are resource agnostic and define a specific characteristic of a subject. These descriptive attributes, such as group membership or role need to be rendered against the applicable resource policy in order to yield access rights. This is typically done through a resource access control or role definition policy.

The advantage of using subject attributes is that the same characteristic can translate into differing access rights depending on the resource at which the request is targeted, as well as the combination of subject attributes the subject holds. The disadvantage is the overhead imposed by the need to issue a subject attribute along with the need to modify resource policies correspondingly to translate the attribute into access permissions. While this may be a small price to pay in larger virtual organizations with long lifetimes and group infrastructure, it is unfeasible for small or ad-hoc groups, especially if the lifetime is short or no group infrastructure exists. In contrast, privilege attributes provide a way of directly assigning access rights on specific resources to subjects without the need to modify and adapt resource policies and thus can support peer-to-peer and ad-hoc collaborative scenarios with less infrastructure required.

Both descriptive and privilege attribute assertions normally make a positive statement, i.e. the holder has the attribute with the following value(s). In theory, one could make a negative statement, i.e., the holder does not have the attribute with the following value(s). Instead, the absence of a positive attribute statement is used. Policy statements could use attributes in either an additive manner, e.g. if the user has the attribute, he has the following rights, or negative, if the user has the attribute, he is denied some rights. The negative case is insecure in a push system, where the initiator may fail to push an attribute that would deny rights. Even in a pull system, the attribute repositories must be in a closed domain, so that no attributes would be missed in a search. Combining policy statements that make both positive and negative assertions about rights is much more complicated than a simple additive scheme. In order to avoid complications and potential policy breeches, it is recommended that subject attributes only grant positive rights and that policy statements are based on the occurrence of an attribute and not the lack of one.

The OGSA Authorization Requirements document [OGSA-Authz-Req] identifies several authorization scenarios that require attribute information be passed between two parties. The first group of scenarios are variants of the push model, in which the initiator retrieves its credentials

from a trusted third party, such as a virtual organization (VO) manager, and passes them to the Grid service controlling access to some resource. Other scenarios are variants of the pull model, in which the initiator passes the Grid service a reference element from which the authorization service (ADF) retrieves the necessary credentials. In order to satisfy the trust relationships between various Grid sites, the assertion must contain sufficient information such that the relying party can determine who made the assertion and that the content was not corrupted in transit. A clean way to solve both of these requirements is to use digitally signed *attribute assertions* that associate an issuer, referred to as a *subject authority* or *privilege authority*, a holder (also referred to as the subject), validity dates and possibly other conditions, with an attribute. If the two parties communicate over an unsecured channel, the issuer must digitally sign each shared assertion. If shared via a secure and authenticated connection, the assertions may be unsigned for efficiency.

An attribute assertion may optionally include some constraints that the issuer wishes to impose on the attribute. These conditions should be simple since they will be combined with any conditions included in the applicable authorization policy. However, they are the only way for the issuer of the attribute to limit its use and validity. Some uses of this feature are to restrict the caching of an attribute, to limit its use to less than a certain level of delegation and to have it take effect only during certain hours of a day.

One of the more obvious requirements of attribute assertions for Grids is the need for extensibility in defining attribute names, values and conditions. On the other hand, in order to allow for the interoperability of different Grid services which enforce authorization (AEF), authorization services (ADF), the attribute issuers, and the policy writers, we need to specify a basic set of elements for attribute assertions and identifiers and values for attributes.

The following section will examine some of the current attribute standards in order to see what is applicable for OGSA.

4. Existing Attribute Standards

4.1 X.509 Attribute Certificate

The IETF PKIX working group defined an X.509 Attribute Certificate that binds attributes to a holder and is digitally signed by an attribute authority. This certificate definition was motivated by the desire to keep attributes out of X.509 public key certificates and encourage the separation of identity and privileges.

The requirements of these certificates include: [RFC3281]

- Issuers of ACs should be able to define their own attribute types for use within closed domains.
- Some standard attribute types, which can be contained within ACs, should be defined. Examples include "access identity," "group," "role," "clearance," "audit identity," and "charging identity."
- Standard attribute types should be defined in a manner that permits an AC verifier to distinguish between uses of the same attribute in different domains. For example, the "Administrators group" as defined by Baltimore and the "Administrators group" as defined by SPYRUS should be easily distinguished.
- It should be possible to "target" an AC at one, or a small number of, servers. This means that a trustworthy non-target server will reject the AC for authorization decisions.

A X.509 Attribute certificate typically has a single subject (called the holder), a number of attributes of possibly varying types. An attribute type is identified by its object identifier [OID] which explicitly refers to a schema definition which defines everything about the attribute, including the name, number and data types of the values and for enumerated types the actual values.

Multiple holders are possible but not widely used. Attribute certificates are ASN.1 encoded, have one validity period and specify the issuer who signed it. They also allow optional extensions that can be used to constrain certificate validity.
The attribute types that have been defined are: *id-aca-authenticationInfo, id-aca-accessIdentity, id-aca-chargingIdentity, id-aca-group, id-at-role, id-at-clearance*. Extensions contain information about the attribute and how to verify it, e.g. revocation locations, keyInfo and audit Identity.

4.2 SAML Attribute Assertions

SAML (Security Assertion Markup Language) defines an XML-based protocol for querying and expressing authentication, attribute and authorization assertions about principals. [SAML] Attribute assertions for a particular subject may be requested via an AttributeQuery wrapped within a SAML request. According to protocol semantics, a SAML response to that request contains zero or more relevant assertions.

SAML also defines an assertion language such that assertions may exist independently to this protocol. Each SAML assertion is a generic packaging of a set of statements pertaining to a particular category (Attribute, AuthorizationDecision or Authentication) into a standard XML structure. Each assertion holds meta-data specific to the assertion itself, such as the issuer identity represented by a string, assertion identifier, and protocol version numbers as well as conditions and advice. Assertion validity dates are a specific form of a condition. Other standard condition definitions address caching and intended audience restrictions. Note that the SAML AuthorizationDecisionStatement is intended to be used in replying to a request for authorization and thus includes the actions that were requested and a Decision with has the values permit, deny or indeterminate. This differs from a privilege attribute which only includes the actions that the subject has with respect to the resource.

As the assertion is the packaging of asserted data, SAML specifies that digital signatures be attached at this level. However, a single SAML assertion can wrap multiple attribute statements. Each attribute statement contains a single subject identity, and one or more attributes, each with zero or more values. Attributes are identified within a statement by an AttributeDesignator. An AttributeDesignator specifies a namespace URI and an attribute name local to that namespace.

4.3 XACML Attributes

XACML (extensible Access Control Markup Language) is designed to express access control policy and the context carried with an initiator when requesting an authorization. [XACML] Both the policy and the request context use attributes. In addition to subject attributes, XACML defines a standard representation for environment, action, and resource attributes. Within an access control policy, an XACML attribute is conceptually specified by a combination of the unique attribute identifier in URI form, a data type and the attribute issuer, and an indicator for its required presence in any context to be evaluated against this policy. This data is defined as an XML complex type named AttributeDesignatorType. The Attribute element is the central abstraction of a request context that will be evaluated against an XACML access control policy. This element comprises meta-data and an attribute value. This meta-data contains the attribute identifier, data-type and issuer so that the ADF may identify any matches with an attribute designator in a policy.

Attributes may be associated with a specific subject in a request context. Further, each subject within a request context may be categorized by the presence of attribute represented by a SubjectAttributeDesignatorType derived from the basic AttributeDesignatorType. XACML defines a number of attribute identifiers for use within a Subject Attribute Designator. They have URIs of the form *urn:oasis:names:tc:xacml:1.0:subject: <id>* and include *subject-id, subject-category, subject-id-qualifier, key-info, authentication-time, authentication-method, request-time, start-time, ip-address*, and *dns-name.* XACML also defines a naming convention to use any identifiers defined in LDAP, e.g., *http://www.ietf.org/rfc/rfc2256.txt#userPassword*.

XACML does not use namespaces for attribute identifiers, does not attach conditions to them, and does not have a specification for signed assertions with validity dates. Instead, the context in which the attribute is embedded may be secured by some means outside of the scope of XACML.

Namespaces were omitted from attributes in order to simplify linking to attributes in policy statements without having a complex format for referencing them. Attributes with the same name in different domains can be named differently to distinguish them, e.g. "permisRole" and "BarcelonaRole". If they have the same format, the same DataType can be associated with both of them.

4.4 Shibboleth

Shibboleth [SHIB], the Internet2 architecture for sharing web resources with access control, defines attributes about its users to the sites. They have extended the XACML naming scheme to include an LDAP schema for eduPerson [EP] that builds on the inetOrgPerson [LDAP]. They specify the names of attributes to be the attributes defined in eduPerson schema, e.g., *eduPersonPrincipalName, eduPersonAffliation,* and *eduPersonExtGroupMembership*.

4.5 EDG VOMS

The European Data Grid has developed a Virtual Organization Membership Service (VOMS) that defines groups, roles and capabilities of its members. [VOMS1] They have specified a Fully Qualified AttributeName (FAQN) that combines these names into attributes for their users. FAQNs are passed within X.509 Attribute Certificates.

4.6 Commonalties and Differences

Below are a number of ways that attribute assertions can be modeled that seem to have similarities between the formats discussed above.

- Number of subjects supported
- Representing multiple values
- Predefined attribute identifiers
- Digital signatures

Below are a number of ways that attribute assertions can be modeled that seem to have differences between the formats discussed above.

- Attribute identifier format
- Attribute meta-data
- Encoding
- Association with a subject or principal

There is typically a single subject who is the holder of one or more attribute(s) (attribute certificates can accommodate multiple holders but its not recommended). A named attribute may have one or more values associated with it. X.509 and SAML can associate conditions with the attribute. Assertion signing is mandatory for X.509 certificates, optional for SAML AttributeAssertions, and not defined in XACML. All three systems allow an attribute to have multiple values. SAML and X.509 allows grouping of several attributes per subject.

5. Standard OGSA Attributes

5.1 Standard Attribute Elements

This section contains a normative specification for the abstract attribute elements.

In order to store attributes in non-secure repositories and to transmit them across unsecured connections, optionally signed attribute assertions are required. The attribute element should be useable in policy statements and should be able to hold environment, action and resource attributes as well as subject and privilege attributes.

These assertions MUST contain the holder of the attribute(s) and one or more attributes. A signed assertion SHOULD contain the identity of the issuer and at least one condition that

contains the validity period of the assertion. All conditions apply to all the attributes. If no validity period is given, the relying party MAY reject the assertion as being invalid. If no issuer is given, the issuer is assumed to be the entity that is securely providing the assertion. The attributes MAY be typed. The attributes MAY be named in a flat name space or MAY have a namespace component. The name and value elements MUST be extensible.

It is also understood that conforming implementations capable of handling more than one system among X.509, SAML and XACML SHOULD be capable to compare for equality issuer and holder names present in any of the systems, when the names themselves are expressed in a comparable format.

```
Attribute Assertion
  Issuer (0 or more)
  Condition (0 or more)
  Holder/Subject (1)
  Attribute (1 or more)
     Name
     Value (0 or more)
     Data Type (0 or 1)
  Signature (optional)
```

5.2 Standard Attribute Types

This section contains a normative specification of attribute names and meanings.

The definitions of attribute identifiers and data types MUST be understood by attribute authorities and policy writers. They MAY need to be understood by an initiator in order to gather up the required attributes before contacting a Grid service. Ideally, attributes can be opaque to the Grid services, authorization services and any attribute repositories. There is a significant class of attributes whose values can be expressed by strings, such as group, role and affiliation. To accommodate these attributes in the simplest way, the ADF will by default use case sensitive string comparison when verifying that an initiator has the required attribute. On the other hand supporting wild-cards in attributes, or non-string values, requires the ADF to understand the data types. For example, the data type field may define a comparison function as XACML does. Also, the data type of environment or initiator context attributes such as IP address or disk quotas, MUST be understood and evaluated by either the AEF or ADF.

It is anticipated that Grid defined attribute names would be defined in a *http://www.gridforum.org/namespaces/2003/06/ogsa-authz/attributeType* namespace and referred to by a URI of the form *http:// www.gridforum.org/namespaces/2003/06/ogsa-authz/attributeType#group.* It is anticipated to follow the Shibboleth and XACML examples and use selected attributes of eduPerson and inetOrgPerson. EduPerson attributes should be given a URI of the form http://www.nmi-edit.org/eduPerson/internet2-mace-dir-eduperson#eduPersonAffliiation. InetOrgPerson attributes should have names of the form http://www.ietf.org/rfc/rfc2256#countryName. It would also help interoperability to accept the XACML request context attributes.

The following attribute names should be used for the stated purposes:

5.2.1 Grid defined subject attribute names

group - an attribute given to a number of individuals to allow a common set of access privileges. Values for groups are completely arbitrary, but might be used for members of a virtual organization, members of an experiment, members of a committee or authors of a paper. Users

may be members of multiple groups. How the rights that a user is granted by multiple groups are combined, is a matter of policy and out of scope for this document.

role - represents some role that an individual may assume for a session. Normally a subject would choose the role or roles for a session and the ADF would not expand those roles. Roles can be hierarchical, where a superior role has all the privileges of an inferior. Examples of Grid roles are: experimenter, administrator, PI. Role hierarchies must be understood by the attribute issuer and the ADF.

fqan – fully qualified attribute name: a compact form to represent both group membership and role ownership. Its format should be <groupname>/Role=<role name>.[VOMS2]

charging-id - a project id or account number to which the current transaction will be charged. If a subject has more than one such identity, then the charging identity must be presented by the subject, unless it can be inferred from the action that is being taken. The values for charging identities will be determined by the party that is charging for the service.

citizenship - Country of which the subject is a citizen. Should name this http://www.ietf.org/rfc/rfc2256#countryName whose values are those given in the ISO 3166-1 list of official short names in English. [ISO 3166-1]

clearance - a security clearance level. The values are defined by the various national agencies or institutions that issue such clearances.

5.2.2 Relevant eduPerson/inetOrgPerson subject attributes

eduPersonAffliliation or **eduPersonPrimaryAffliliation** - Specifies the person's relationship(s) to the institution in broad category: Values may be faculty, student, staff, alum, member, affiliate, employee. One of the purposes of this value is to indicate sets of privileges that go with certain relationships.

eduPersonEntitlement - a URI that indicates a set of rights to specific resources. The value of this URI is a contract or agreement name. The rights that this contract allows are determined out-of-band between the provider and the licensee. Note that this is not considered a privilege attribute since the actual rights are not specified in the attribute. There is still a level of indirection where the ADF must know or discover what right the URI grants.

eduPersonOrgDN - the DN of the directory entry representing the institution with which the person is associated. EduPerson has added this, instead of just using the orgPerson Organization attribute, to facilitate the discovery of more information about the organization.

eduPersonOrgUnitDN - same as OrgDN only for the person's organization unit. Note having these assigned as attributes outside of the components of a user's name would solve the problem that this information is not contained in Grid DNs.

eduPersonPrincipalName - this is a name of the form user@univ.edu where univ.edu is the local security domain. The user name must be unique within the domain and the user should be able to authenticate locally with this name. It may be implemented as a Kerberos identifier or as an email address. It is intended to be used to support systems that do not use PKI and may eventually be deprecated.

5.2.3 Grid defined privilege names

Currently only one standard privilege attribute is defined. Applications might want to define additional privilege attributes such as file-privilege, network-privilege or computational-privilege. Until more experience is gained using privilege attributes, the value will be specified as containing both the rights and the resource to which they apply. The syntax of the value must be agreed upon between the Attribute Authority that issues them and the AEF that uses them.

privilege – the value contains the rights and the resource to which they apply. The value is variable length UTF-8 String whose content must be agreed upon by the issuer and the ADF.

Two possibilities for representing privileges are an XACML rule construct that consists of a subject, resources and actions, or a resource URI followed by a comma separated list of rights, e.g. PrivilegeType://resource,read,write. One problem with the XACML rule is that the subject must either match the holder of the attribute assertion or be set to <AnySubject/>.

If a privilege attribute is not understood by a relying party it may be silently ignored. Thus complying implementations MUST only use PrivilegeAttributes that are positive. Negative privileges (deny rules) are not permitted. The use of negative privileges would lead to security breaches when privileges are omitted, missing or not understood.

5.3 Standard conditions

This section contains a normative specification of attribute assertion conditions and their meanings. A condition applies to all the attributes contained in an attribute assertion. Thus attributes requiring different conditions must be packaged in different assertions.

An attribute authority constrains the use of an attribute via conditions. Conditions SHOULD be kept simple because if a relying party does not understand how to process the condition, it MUST not use the attribute. We need to support single value conditions, like DoNotCache, conditions that are equal to one or more values, e.g. audienceRestriction and conditions that are expressed by algebraic expressions combining terms and values that are known by the policy writers, the authorization service and the Grid service (AEF). Some generally useful conditions on attributes are:

- AudienceRestriction or Target – restrict the use of the attribute to only some target resources
- Time of day, e.g. time >= 8:00 & time <= 17:00
- Days of week, e.g. day != sat & day != sun
- Making one attribute depend on the existence of another e.g role=administrator if project=Atlas

The standard vocabulary for such expressions includes the relational operators: =, !=, <, >, <=, >=, &, |, times of day hh:mm with a 24 hr clock, days of the week: sun-sat. It is recommended that we use the XACML functions and format for relational expressions. They have the advantage of not using symbols such as <, >, & that require escaping in XML expressions, and may also allow code reuse of libraries developed for XACML ADFs. They are defined in the XACML v1.1 document, starting on page 100. E.g. *String-equal, integer-equal, boolean-equal, date-equal, time-equal, x500name-equal, string-greater-than, string-greater-than-or equal,* etc..

6. SAML profile for attribute assertions

This section contains a normative specification of how the attributes and conditions defined in the previous sections should be expressed using SAML. This document does not require the use of SAML for expressing Attribute Assertions in OGSA, but only defines how it MUST be used if chosen by the implementer.

The SAML Assertion element is used by one entity to assert the statements about a principal. While an Assertion element can contain a variety of SAML statements, for the purposes of this document we consider only AttributeStatements. The Assertion element includes the following elements:

- An optional *Conditions* element specifying the conditions for use of the assertion.
- An optional *Advice* element specifying advice for use of the element.
- Zero or more *AttributeStatements* specifying attributes.

- An optional *Signature* element allowing the Assertion to be verified.

It also carries the following information as XML attributes:

- The issuer (the attribute authority)
- The issue instant (date/time)

The following subsections describe the use and extensions to these elements for OGSA.

6.1 Conditions Element

Implementations are advised to be conservative in their use of this element and only include it when they are confident it will be understood. Relying parties MUST not use an attribute if they do not understand how to evaluate any of its conditions. Implementations MAY support only standard conditions.

The Conditions element can contain optional time constraints and/or zero or more Condition elements (note difference in plurality between element names) on the assertion. Several basic condition types, such as cache behavior or audience restrictions, are directly defined in the specification [SAML] as well as an abstract condition element that serves as an extension point. These extended conditions should be used to express particular constraints that the attribute authority wishes to place on the use of the attribute by the subject. One of the most obvious uses for this is to limit the time of day that a subject can act in a specific role.

6.2 Advice Element

This specification recommends against the use of the Advice element. Implementations SHOULD NOT use this element and MAY only include it when they are confident that it will be understood.

6.3 AttributeStatement Element

The Attribute Statement contains the following elements:

- Subject element
- One or more attributes consisting of
    - Attribute name and name space
    - One or more attribute values

When the assertion encapsulating the Attribute Statement is passed across an insecure network, it MUST be signed by the attribute authority.

6.3.1 Subject Element

This element contains the name of the attribute holder. The Subject and contained NameIdentifer elements are unchanged from the SAML specification. The exact use of these elements is driven by the authentication mechanism used by the client. In some scenarios, the authorization service (ADF) MAY require the holder and client names to be the same. In other scenarios, the authorization service MAY allow trusted clients to request authorization decisions on behalf of any initiator.

The SAML specification defines how some common identity types are asserted. The Grid Security Infrastructure (GSI) is a common Grid authentication mechanism that uses X.509 based identities. The SAML specification defines a URI for X.509 subject names (#X509SubjectName) that SHOULD be used for GSI authenticated identities. Note that SAML specifies the LDAP encoding of DNs [RFC2253].

6.4 Signature Element

This specification places no constrains on the Signature elements. Implementations MUST sign assertions when they do not have an authenticated and secure connection to the evaluator of the assertion.

7. X.509 Attribute Certificate profile for OGSA attribute assertions

This section presents a normative profile for X.509 Attribute Certificates that convey OGSA attribute assertions, including a set of fundamental attributes and their ASN.1 encoding for the grid community. It is possible that the same holder is known to the issuer with many different names and credentials. In this case, the name used for the holder field of the AC should be the same one present in the credentials with which the holder was authenticated, and this same credential should be the one considered during the verification phase of the AC.

AC Required Contents:

Version number – Version 2

Holder may be one of the following three types:

- A general name object holding a X500Name (Directory Name)
  This is the RECOMMENDED way to identify the holder if the attribute SHOULD be used in an environment where a holders is identified by a X500Name and the attribute cannot be bound to a specific public-key certificate (PKC) of the holder.
- baseCertID
  This holder identification is RECOMMENDED if the attribute is used in a context where a holder was authenticated using a PKC and the loss in flexibility due to the binding to a specific certificate as well as the implied coupling of the AC lifetime to the PKC lifetime does not pose a problem. In addition this option may provided added privacy since the holder's name is not visible.
- objectdigestinfo
  This holder type is useful for software objects (it can be a hash of the code) and can also be a public key id, thereby not requiring public-key certificates for the identity

Issuer - A general name object holding a X500Name (Directory Name)

Signature - Algorithm identifier used to create the hash for the AC signature.

Serial number – a unique identifier suitable for use in revocation as defined in RFC3281

Validity period – as defined in RFC3281

Attributes – 1 or more attribute object(s) as defined in RFC3281

Extensions – 0 or more extensions are permitted

NoRevocation

extension should be present if no CRLs are issued

CRLDP

should be present if CRLs are to be issued

BasicAttConstraints

should be optional, with authority set to FALSE if present and the extension set to non-critical. Thus is can be ignored for now, but is there in preparation for delegation of authority which we all know we will need in the longer term. It will allow software creators to start to migrate towards dynamic delegation.

AuthorityAttributeIdentifier

won't be needed for now, but should be mandatory once dynamic delegation is added. (this extension is a back pointer to the issuing AA's AC from the holder's AC)

AttributeConditions

format + OID need to be defined here

Targets

can be used to restrict this AC only to a selected list targets. These targets should be specified with a fully qualified domain name.

HolderUniqueIdentifier UTF8STRING OID 1.3.6.1.4.1.8005.100.100.7

This extension, if present, SHOULD contain an additional name of the holder that the Attribute Issuer guarantees to remain valid for as long s it know the Holder

Security Considerations

This specification defines the elements and use of attributes for authorization services. Implementers of attributes need to be aware that errors in implementation could lead to denial of service or improper granting of service to unauthorized users. Users of attribute assertions should be aware of the situations in which they must require and verify signed assertions.

Author Information

Mary R. Thompson
Lawrence Berkeley National Laboratory
MRThompson@lbl.gov

Von Welch
University of Chicago
welch@mcs.anl.gov

Markus Lorch
Department of Computer Science
Virginia Tech
mlorch@vt.edu

Rebekah Lepro
NASA, Ames
bekah@nas.nasa.gov

David Chadwick
Information Systems Institute
University of Salford
d.w.Chadwick@salford.ac.uk

Vincenzo Ciaschini
INFN - CNAF
Viale Berti Pichat, 6/2
I - 40127 BOLOGNA
vincenzo.ciaschini@cnaf.infn.it

Glossary

The following terms are abbreviations are used in this document.

AA – Attribute Authority, Principal that is trusted to issue attribute assertions.

ACI – Access Control Information (from ISO 10181-3). Any information used for access control purposes, including contextual information.

ADF – Access control Decision Function (from ISO 10181-3). A specialized function that makes access control decisions by applying access control policy rules to an access request, ADI (of initiators, targets, access requests, or that retained from prior decisions), and the context in which the access request is made.

ADI – Access control Decision Information (from ISO 10181-3). The portion (possibly all) of the ACI made available to the ADF in making a particular access control decision.

AEF – Access control Enforcement Function (from ISO 10181-3). A specialized function that is part of the access path between an initiator and a target on each access request and enforces the decision made by the ADF.

Client – the entity making a decision request to the ADF (it could be the target, the initiator, or a proxy acting on behalf of the initiator)

Contextual information – Information about or derived from the context in which an access request is made (e.g. time of day).

Descriptive Attribute - An attribute assigned to an entity by an authority that describes a characteristic of that entity. Sample attributes are roles held by an entity or accounting information associated with an entity. Descriptive attributes usually yield access rights indirectly after being rendered against applicable access control policies.

Environmental parameters – same as contextual information.

Initiator – An entity (e.g. human user or computer-based entity) that attempts to access other entities (from ISO 10181-3).

OID - Object Identifier, a strings of numbers allocated in a hierarchical manner, so that, for instance, the authority for "1.2.3" is the only one that can say what "1.2.3.4" means. The formal definition of OIDs comes from ITU-T recommendation X.208 (ASN.1). OIDs are assigned by the Internet Assigned Numbers Authority (IANA)

PDP – Policy Decision Point (from RFC2904), same as ADF

PEP – Policy Enforcement Point, (from RFC2904) same as AEF

Privilege Attribute - An attribute assigned to an entity by an authority that describes a distinct direct access right that the holding entity has on a (set of) specific resource object(s).

Relying party - The entity that uses information such as attribute assertions, or authorization assertions to allow some actions.

Resource Attribute – A descriptive attribute bound to a resource (e.g. a security clearance a resource has).

Subject - same as initiator (used by SAML and XACML)

Subject Attribute – A descriptive attribute bound to a subject (typically a user).

Target – An entity, usually a resource, to which access may be attempted (from ISO 10181-3).

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be

available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Full Copyright Notice

References

[Akenti] Thompson, M., et al., "Certificate-based Access Control for Widely Distributed Resources," in Proc. 8th Usenix Security Symposium. 1999.

[Authz] Welch, V., et al, OGSA Authorization Requirments, June, 2003.

[CAS] Pearlman, L., V. Welch, I. Foster, C. Kesselman, S. Tuecke, "A Community Authorization Service for Group Collaboration," Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

[EP] “Class Object Specification 200210”, http://www.educause.edu/eduperson/

[ISO 3166-1] http://www.iso.ch/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html

[LDAP] Smith, M., “Definition of the inetOrgPerson”, RFC2787, April 2000

[OGSA-Authz-Req] Welch, V., Siebenlist, F., Chadwick, D., Meder, S., Pearlman, L. “OGSA Requirements” June 2003 https://forge.gridforum.org/docman2/ViewCategory.php?group_id=119&category_id=449

[OGSA-Service] Welch, V., Siebenlist, F., Chadwick, D., Meder, S., Pearlman, L. “Use of SAML for OGSA Authorization”, Sept 2003 https://forge.gridforum.org/docman2/ViewCategory.php?group_id=119&category_id=450

[OGSI] Foster, I., C. Kesselman, J. Nick, S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[PERMIS] Chadwick, D.W., Otenko, O.," The PERMIS X.509 Role Based Privilege Management Infrastructure", Proceedings of 7th ACM Symoisium on Access Control Models and Technologies (SACMAT 2002).

[Roadmap] Siebenlist, F., et al, "OGSA Security Roadmap," OGSA Security WG, Global Grid Forum, July, 2002.

[RFC2904] Vollbrecht, J., et al, " AAA Authorization Framework", RFC 2904, August 2000.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997.

[RFC2253] Wahl, M., Kille, S., Howes, T., " Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", Dec. 1977

[RFC3281] Farrell, S., Housley, R. "An Internet Attribute Certificate Profile for Authorization", RFC 3281, May 2002.

[SAML] "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1, July 13, 2003, OASIS Security Services Technical Committee, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security, June, 2003.

[SHIB] Erdos, M., Cantor, S., "Shibboleth-Architecture DRAFT v0.5, http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-architecutre-05.pdf

[VOMS1] "VOMS Architecture v1.1," http://grid-auth.infn.it/docs/VOMS-v1_1.pdf, February 2003.

[VOMS2] "VOMS Credential Format" http://forge.gridforum.org/projects/ogsa-auth/document/voms-credential-format/en/1

[XACML] "OASIS eXtensible Access Control Markup Lanugage (XACML) Committee specification 1.0", Dec. 2002,OASIS Security Services Technical Committee, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml, Feb, 2003.