Kerberos SAML Solution Profile & Bindings

Working Draft 04, 15th March 2004

Document identifier:
draft-sstc-solution-profile-kerberos-04

Location:
http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

Editors:
Tim Alsop, CyberSafe Limited (tim.alsop@cybersafe.ltd.uk)
John Hughes, Entegrity Solutions (john.hughes@entegrity.com)

Contributors:
Scott Cantor, Individual
Jeff Hodges, Sun Microsystems
Ron Monzillo, Sun Microsystems

Abstract:
This document describes the profiles and bindings for using the Kerberos protocol with SAML to provide a Single Sign-On ("SSO") service to users and applications, and/or provide integration with an existing Kerberos authentication infrastructure that might be deployed.

Status:
Interim draft. Please send comments to the editors.

Committee members should send comments on this specification to the securityservices@lists.oasis-open.org list. Others should subscribe to and send comments to the security-services-comment@lists.oasis-open.org list. To subscribe, send an email message to security-services-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Security Services TC web page (http://www.oasisopen.org/committees/security/).

Table of Contents

1Introduction....4
1.1Terminology....4
2Use Cases....5
2.1Use Case – Kerberos Client....5
2.2Use Case – Kerberos Server....5
3Profiles....7
3.1Approach....7
3.2Browser Kerberos Profiles....7
3.3Browser/Artifact Kerberos Profile (BAKP) of SAML....8
3.3.1Required Information....8
3.3.2Preliminaries....8
3.3.3Step 1: Accessing the Inter-Site Transfer Service....8
3.3.4Step 2: Return of Artifact....9
3.3.5Step 3: Accessing the Artifact Receiver URL....9
3.3.6Steps 4 and 5: Acquiring the Corresponding Assertions....9
3.3.7Step 6: Responding to the User's Request for a Resource....10
3.3.8Artifact Format....10
3.3.9Overall Processing....10
3.4Browser/POST Kerberos Profile (BPKP) of SAML....12
3.4.1Required Information....12
3.4.2Preliminaries....12
3.4.3Step 1: Accessing the Inter-Site Transfer Service....13
3.4.4Step 2: Return of the Assertion....13
3.4.5Step 3: Posting the Form Containing the Response....13
3.4.6Step 4: Responding to the User's Request for a Resource....14
3.4.7Overall Processing....14
4Bindings....17
4.1Introduction....17
4.2Browsers, Web Server's and Kerberos....17
4.2.1Kerberos Client at Web Server....17
4.2.2Kerberos Client at Workstation or Browser....17
4.3Kerberos Bindings for SAML 2.0....18
4.3.1GSSAPI with Kerberos bindings....18
4.3.2GSSAPI profile of SASL bindings....19
5Normalization and SAML Identifiers....20
5.1Authentication Method Identifiers....20
5.1.1Kerberos....20
5.2NameIdentifier Format Identifiers....20
5.2.1Kerberos Principal Name....20
5.3Kerberos Attributes and Naming....20
5.4Microsoft Windows Kerberos....21
5.5Distributed Computing Environment (DCE)....21
5.6Kerberos Service Ticket....21

6References.....22
6.1Normative References.....22
7Notices.....25

1 Introduction

This document explains how the Kerberos protocol can be used in conjunction with SAML in order to :

1. Provide a secure and trusted mechanism to pass a user identity to the SAML Authentication Authority so that an artifact or assertion can be returned using the authenticated identity of the user;
2. Implement a Single Sign-On (“SSO”) experience for users - especially useful when the workstation and/or server operating systems have a Kerberos implementation available and multiple vendors operating systems are used;

The various implementations of Kerberos are catered for in this document, in particular :

- An implementation based on the Kerberos standard, as defined in [RFC 1510];
- A DCE (Distributed Computing Environment) based implementation;
- A deployment of Microsoft Kerberos, as implemented in Windows 2000, XP and 2003.

1.1 Terminology

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 IETF [RFC 2119].

2 Use Cases

(This section is intended to go into the SAML 2.0 Technical Overview)

Two primary use cases are defined for use when combining Kerberos and SAML. They are described in the next two sections.

2.1 Use Case – Kerberos Client

This use case has a user of a client workstation authenticating with the local site (the Identity Provider), using Kerberos. Following successful authentication the user's initial Kerberos credentials reside on the workstation in a memory resident credential cache.

The workstation user then wishes to gain access to resources on a remote site in another management domain, so that:

- No further authentication is required;
- Session attributes are transferred seamlessly over to the remote application, via a SAML assertion, so that it can make appropriate authorization decisions;
- The remote site does not need any Kerberos protocol support to recognize, or verify the user's identity.

Figure 1 illustrates the high level use case.

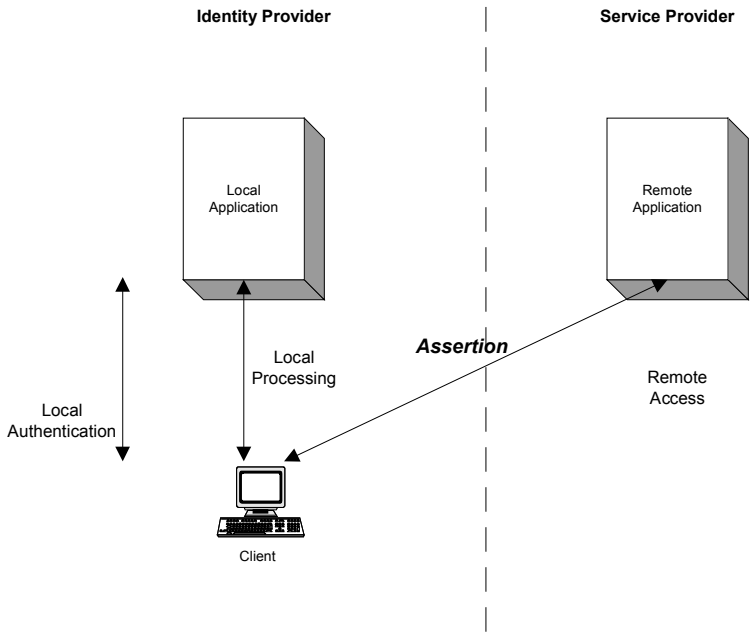


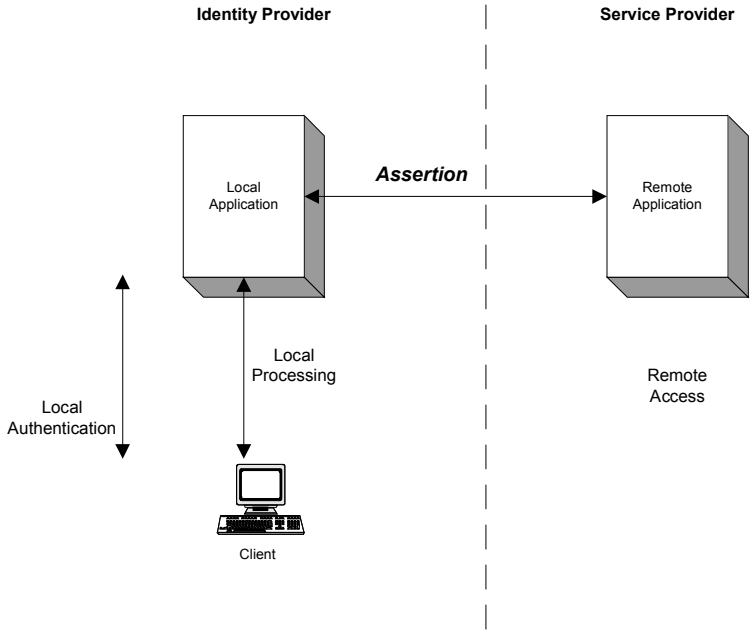
Figure 1 – Use Case – Kerberos Client

2.2 Use Case – Kerberos Server

This use case differs from the previous one, in that the Kerberos credentials are stored in the local application. The authentication technology used to authenticate the user to the local application is not defined. This use case represent the situation when two applications need to communicate securely and the local application has a requirement to pass appropriate attributes securely to a remote application.

123

124 Figure 2 illustrates the high level use case.



125 Figure 2 – Use Case – Kerberos Server

126

3 Profiles

(Parts of this section is intended to go into the SAML 2.0 Technical Overview and also the SAML 2.0 Profiles document)

3.1 Approach

This document approaches the need for SAML and Kerberos to co-existence from three different aspects:

- How Kerberos can be used as the security technology to secure SAML requests and responses. The various techniques available are described in the Bindings section of this document
- How Kerberos identity information is mapped into the various elements and values supported by the `<Subject>` element within a SAML assertion. This is described in the *Normalization and SAML Identifiers* section of the document
- How Kerberos Service Tickets can be used as Subject Confirmation Data by the Service Provider. This is described in the *Normalization and SAML Identifiers* section of the document

Whilst other application domain profiles can be defined, in particular interest is the use of Kerberos in a Web Browser environment. Therefore the two normative Profiles provided define how Kerberos can be used in conjunction with the existing SAML 1.x Browser/Artifact and Browser/POST profiles. In particular the aim is to re-use as much as possible of the infrastructure used to support the two Browser SSO profiles defined in SAML 1.x. The following two sections define the profiles

3.2 Browser Kerberos Profiles

In the scenario supported by the web browser Kerberos profiles, a web user authenticates to a *source site* using Kerberos. The web user then accesses a secured resource at a destination site, without directly authenticating to the *destination site*.

The following assumptions are made about this scenario for the purposes of these profiles:

- The user is using a standard commercial browser and has authenticated to a source site using Kerberos. The user's Kerberos credentials are stored on the workstation in a memory resident Kerberos credentials cache.
- The workstation has Kerberos client code resident on it

At some point, the user attempts to access a *target* resource available from the destination site, and subsequently, through one or more steps, arrives at an *inter-site transfer service* (which may be associated with one or more URIs) at the source site. Starting from this point, the web browser Kerberos profiles describe a canonical sequence of exchanges that transfer the user browser to an *assertion consumer service* at the destination site. Information about the SAML assertions provided by the source site and associated with the user, and the desired target, is conveyed from the source to the destination site by the protocol exchange.

As with the Web Browser SSO Profiles two techniques are defined, based upon the Browser/Artifact SSO Profile and the Browser/POST SSO Profile:

- **SAML Artifact:** A SAML artifact of “small” bounded size is carried to the destination site as part of a URL query string such that, when the artifact is later conveyed back to the source site, the artifact unambiguously references an assertion. The artifact is conveyed to the destination site, which then acquires the referenced assertion from the source site by some further steps. Typically, this involves the use of a registered SAML protocol binding. This technique is used in the browser/artifact Kerberos profile of SAML.
- **Form POST:** SAML assertions are uploaded to the browser within an HTML form and conveyed to the destination site as part of an HTTP POST payload when the user submits the form. This technique is used in the browser/POST Kerberos profile of SAML.

3.3 Browser/Artifact Kerberos Profile (BAKP) of SAML

3.3.1 Required Information

Identification: urn:oasis:names:tc:SAML:2.0:profiles:kerberos-artifact-01

Contact information: security-services-comment@lists.oasis-open.org

SAML Confirmation Method Identifiers: The "SAML artifact" confirmation method identifier is used by this profile. The following RECOMMENDED identifier has been assigned to this confirmation method:

urn:oasis:names:tc:SAML:1.0:cm-artifact

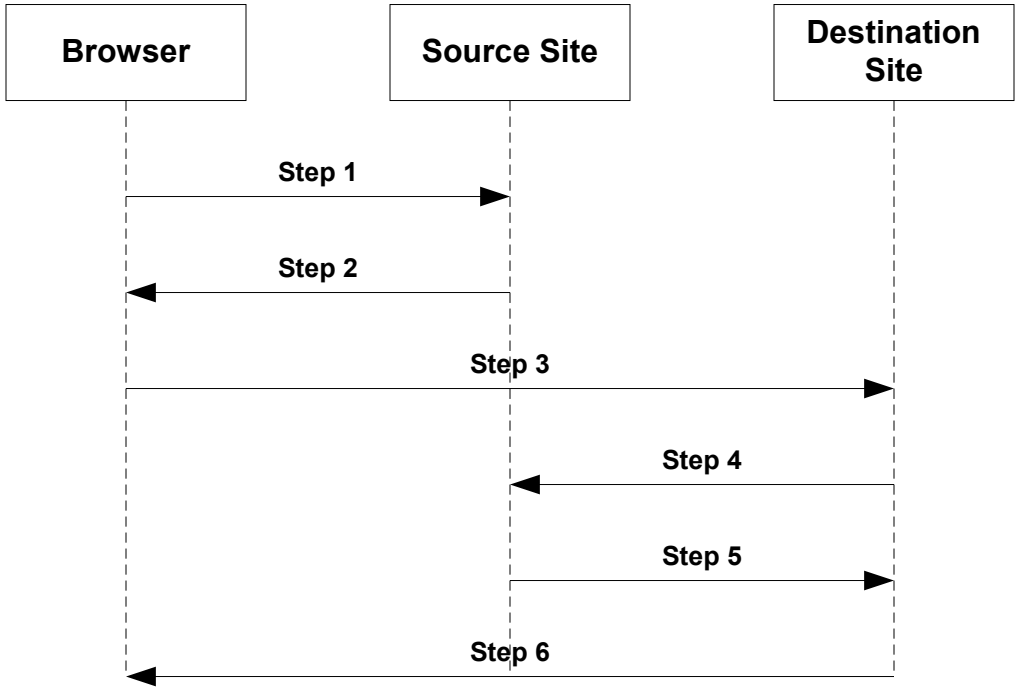
Description: Given below.

Updates: None.

3.3.2 Preliminaries

The browser/artifact Kerberos profile of SAML relies on a reference to the needed assertion traveling in a SAML artifact, which the destination site must dereference from the source site in order to determine whether the user is authenticated.

The browser/artifact Kerberos profile consists of a single interaction among three parties (a user equipped with a browser, a source site, and a destination site), with a nested sub-interaction between two parties (the source site and the destination site). The interaction sequence is shown in the following figure, with the following sections elucidating each step.



3.3.3 Step 1: Accessing the Inter-Site Transfer Service

In step 1, the user's workstation accesses the inter-site transfer service at the source site sending to it an `<AuthnRequest>`. The set of RECOMMENDED Kerberos protocol binding is defined in the binding section. The ITS determines the identity of the subject from the provided Kerberos Principal.

Note that unlike the Browser/Artifact SSO profile the workstation does not provide the TARGET, this profile assumes that the workstation knows this value. Refer to the non-normative overall processing section for a description of how the TARGET value can be obtained by the workstation.

3.3.4 Step 2: Return of Artifact

In step 2, the source site's inter-site transfer service responds and returns to the workstation the artifact. The protocol binding is the same as used in step 1.

3.3.5 Step 3: Accessing the Artifact Receiver URL

In step 3, the user's workstation accesses the artifact receiver service at host https://<artifact receiver host name>, with a SAML artifact representing the user's authentication information attached to the URL. This step is identical to that in the Browser/Artifact SSO profile.

The HTTP request MUST take the form:

```
GET <path>?…<SAML searchpart>…<HTTP-Version>
<other HTTP 1.0 or 1.1 request components>
```

Where:

<artifact receiver host name>

This provides the host name and optional port number at the destination site where the artifact receiver service URL associated with the assertion consumer service is available.

<path>

This provides the path components of the artifact receiver service URL at the destination site.

<SAML searchpart>= …**TARGET=<Target>**…**SAMLart=<SAML artifact>** …

A single target description MUST be included in the `<SAML searchpart>` component. At least one SAML artifact MUST be included in the `<SAML searchpart>` component; multiple SAML artifacts MAY be included. If more than one artifact is carried within `<SAML searchpart>`, all the artifacts MUST have the same `SourceID`.

Confidentiality and message integrity MUST be maintained in step 3. It is RECOMMENDED that the artifact receiver URL be protected by SSL 3.0 or TLS 1.0 (see Section). Otherwise, the artifacts transmitted in step 3 will be available in plain text to any attacker who might then be able to impersonate the assertion subject.

3.3.6 Steps 4 and 5: Acquiring the Corresponding Assertions

In steps 4 and 5, the destination site, in effect, dereferences the one or more SAML artifacts in its possession in order to acquire a SAML assertion that corresponds to each artifact.

These steps MUST utilize a SAML protocol binding for a SAML request-response message exchange between the destination and source sites. The destination site functions as a SAML requester and the source site functions as a SAML responder.

The destination site MUST send a `<samlp:Request>` message to the source site, requesting assertions by supplying assertion artifacts in the `<samlp:AssertionArtifact>` element.

If the source site is able to find or construct the requested assertions, it responds with a `<samlp:Response>` message with the requested assertions. Otherwise, it responds with a `<samlp:Response>` message with no assertions. The `<samlp:Status>` element of the `<samlp:Response>` MUST include a `<samlp:StatusCode>` element with the value `Success`.

In the case where the source site returns assertions within `<samlp:Response>`, it MUST return exactly one assertion for each SAML artifact found in the corresponding `<samlp:Request>` element. The case where fewer or greater number of assertions is returned within the `<samlp:Response>` element MUST be treated as an error state by the destination site.

The source site MUST implement a "one-time request" property for each SAML artifact. Many simple implementations meet this constraint by an action such as deleting the relevant assertion from persistent storage at the source site after one lookup. If a SAML artifact is presented to the source site again, the source site MUST return the same message as it would if it were queried with an unknown artifact.

The selected SAML protocol binding MUST provide confidentiality, message integrity, and bilateral authentication. The source site MUST implement the SAML SOAP binding with support for confidentiality, message integrity, and bilateral authentication.

The source site MUST return a response with no assertions if it receives a `<samlp:Request>` message from an authenticated destination site *X* containing an artifact issued by the source site to some other destination site *Y*, where *X* <>*Y*. One way to implement this feature is to have source sites maintain a list of artifact and destination site pairs. The `<samlp:Status>` element of the `<samlp:Response>` MUST include a `<samlp:StatusCode>` element with the value `Success`.

At least one of the SAML assertions returned to the destination site MUST be an *SSO assertion*.

Authentication statements MAY be distributed across more than one returned assertion.

Every subject-based statement in the assertion(s) returned to the destination site MUST contain a `<saml:SubjectConfirmation>` element as follows:

- The `<saml:ConfirmationMethod>` element MUST be set to urn:oasis:names:tc:SAML:1.0:cm:artifact.
- The `<SubjectConfirmationData>` element SHOULD NOT be specified.

Based on the information obtained in the assertions retrieved by the destination site, the destination site MAY engage in additional SAML message exchanges with the source site.

3.3.7 Step 6: Responding to the User's Request for a Resource

In step 6, the user's browser is sent an HTTP response that either allows or denies access to the desired resource.

No normative form is mandated for the HTTP response. The destination site SHOULD provide some form of helpful error message in the case where access to resources at that site is disallowed.

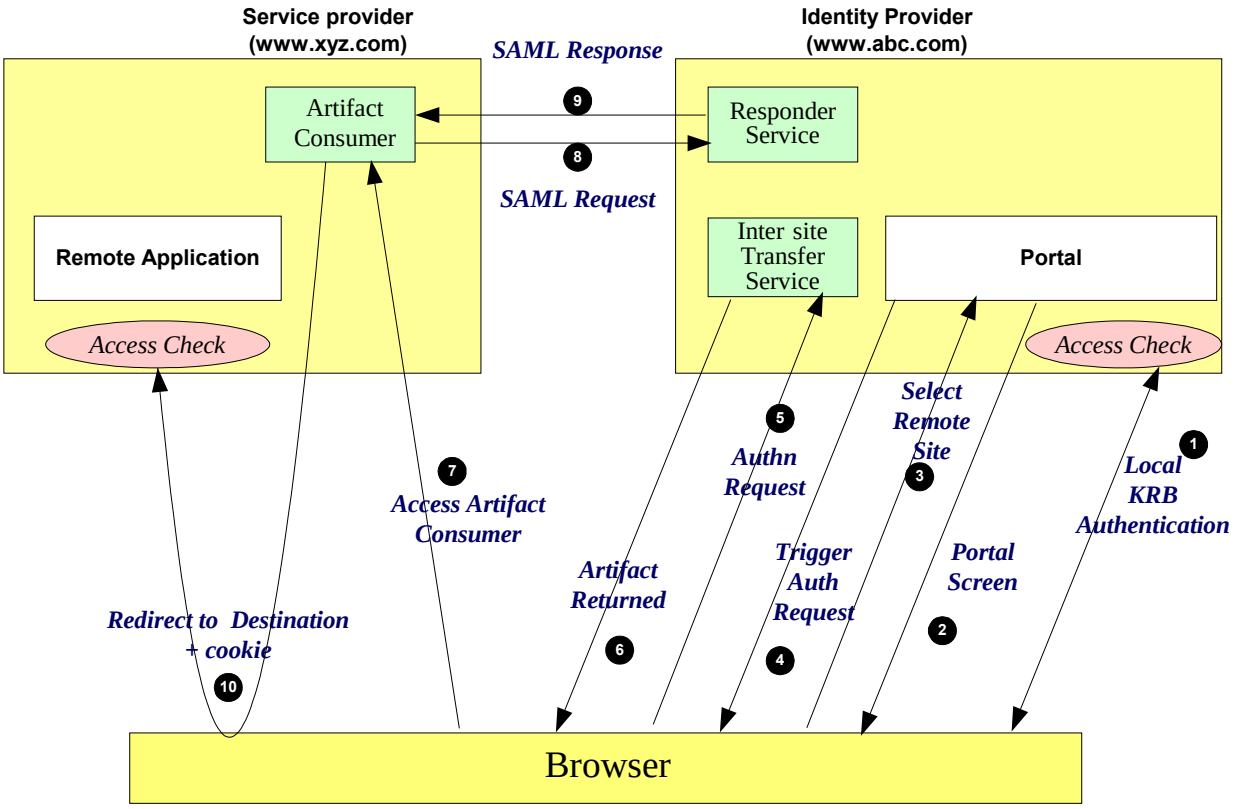
3.3.8 Artifact Format

The Artifact format is the same as defined for the Browser/Artifact SSO profile.

3.3.9 Overall Processing

The following figure shows the processing and message flows for the Browser/Artifact Kerberos profile in the Local-Site-First scenario. In this example, the local web site includes a component called an Inter-site Transfer Service (ITS).

The workstation contains a Kerberos client with additional SAML functionality. The example describes the use of a Java applet on the workstation to perform part of the processing, however any other suitable technology could be used. It is highly RECOMMENDED that if a Java applet is used then it is loaded locally from the workstation file store.



The processing is as follows:

1. The user authenticates to the source site using Kerberos.
2. At some point the user wishes to use web based applications and may access a "Portal" web page listing resources that they may access.
3. The user selects a menu option (or function) on the Portal web page that means the user wants to access a resource or application on a destination web site www.xyz.com (although, of course, the user may not be made aware of this).
4. The portal application then sends a HTML page to the workstation that causes an applet to be executed. The HTML contains the URL of the resource on the remote site. This is known as the TARGET URL. The HTML also contains binding information for the source site Inter-Site transfer Service. The binding value provided will depend on the binding being used, it will have to uniquely define the location of the ITS on the source site. The HTML page also contains the URL of the destination site's Artifact Consumer service. The HTML page would take the form of:

```
<title>A Kerberos/SAML client</title>
<hr>
<applet code "kerberosSaml.class">
<param name=TARGET value="http://www.xyz.com/index.asp">
<param name=ITS value={binding}>
<param name=CONSUMER value="http://www.xyz.com:8001/ArtifactConsumer
</applet>
<hr>
```

5. The workstation application (e.g. the Java applet) sends to the Inter-Site Transfer service an `<AuthnRequest>` using an appropriate binding protocol (as described in the bindings section). The ITS uses the Principal Name from Kerberos to define the Subject.
6. The Inter-site Transfer Service generates an assertion for the Subject while also creating an artifact.

The artifact contains the source ID of the www.abc.com SAML responder together with a reference to the assertion (the AssertionHandle). The raw artifact is sent back to the workstation using the same protocol binding used in Step 5.

7. The workstation issues a HTTP GET request to the Artifact Consumer providing the artifact and TARGET as query variables., for example:

   https://www.xyz.com:7001/ArtifactConsumer?TARGET=http://www.xyz.com/index.asp&SAMLart=<artifact>

8. On receiving the HTTP message, the Artifact Receiver, on the remote web site, extracts the source-ID. A mapping between source IDs and remote Responders will already have been established administratively. The Artifact Receiver will therefore know that it has to contact the www.abc.com SAML responder at the prescribed URL. The www.xyz.com Artifact Receiver will send a SAML request to the www.abc.com SAML responder containing the artifact supplied by the Inter-site Transfer Service of www.abc.com.
9. The www.abc.com SAML responder supplies back a SAML response message containing the assertion generated during step 7. In most implementations, if a valid assertion is received back, then a session on www.xyz.com is established for the user (the relying party) at this point.
10. The Artifact Receiver, on the remote web site, sends a redirection message containing a cookie back to the browser. The cookie identifies the session. The browser then processes the redirect message and issues a HTTP GET to the TARGET resource on www.xyz.com. The GET message contains the cookie supplied back by the Artifact Receiver. An access check is then back to established whether the user has the correct authorization to access the www.xyz.com web site and the index.asp resource.

3.4 Browser/POST Kerberos Profile (BPKP) of SAML

3.4.1 Required Information

Identification: urn:oasis:names:tc:SAML:2.0:profiles:kerberos-post-01

Contact information: security-services-comment@lists.oasis-open.org

SAML Confirmation Method Identifiers: The "Bearer" confirmation method identifier is used by this profile. The following identifier has been assigned to this confirmation method:

urn:oasis:names:tc:SAML:1.0:cm:bearer

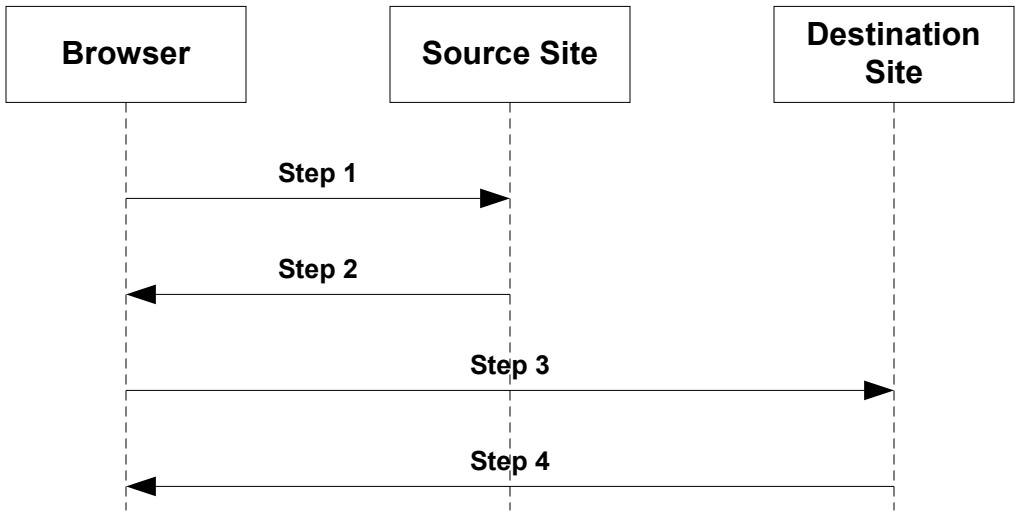
Description: Given below.

Updates: None.

3.4.2 Preliminaries

The browser/POST Kerberos profile of SAML allows authentication information to be supplied to a destination site without the use of an artifact. The following figure diagrams the interactions between parties in the browser/POST Kerberos profile.

The browser/POST Kerberos profile consists of a series of two interactions, the first between a user equipped with a browser and a source site, and the second directly between the user and the destination site. The interaction sequence is shown in the following figure, with the following sections elucidating each step.



3.4.3 Step 1: Accessing the Inter-Site Transfer Service

In step 1, the user's workstation accesses the inter-site transfer service at the source site sending to it an <AuthnRequest>. The set of RECOMMENDED Kerberos protocol binding is defined in the binding section. The ITS determines the identity of the subject from the provided Kerberos Principal.

Note that unlike the Browser/POST SSO profile the workstation does not provide the TARGET during the stage, this profile assumes that the workstation knows this value from a previous interaction. Refer to the non-normative overall processing section for a description of how the TARGET value can be obtained by the workstation.

3.4.4 Step 2: Return of the Assertion

In step 2, the source site's inter-site transfer service responds and returns to the workstation a SAML Response containing the Assertion. The SAML: Response is returning using the same session as set up in Step 1, hence using the same protocol binding.

3.4.5 Step 3: Posting the Form Containing the Response

In step 3, the workstation submits a form containing the SAML response using the following HTTP request to the assertion consumer service at host https://<assertion consumer host name>. The workstation performs the wrapping of the SAML response within the HTTP Form.

The HTTP request MUST include the following components:

```
POST <path> <HTTP-Version>
<other HTTP 1.0 or 1.1 request components>
```

Where:

<assertion consumer host name>

This provides the host name and optional port number at the destination site where the assertion consumer service URL is available.

<path>

This provides the path components of the assertion consumer service URL at the destination site.

<other HTTP 1.0 or 1.1 request components>

This consists of the form data set derived by the browser processing of the form data received in step 2 according. Exactly one SAML response MUST be included within the form data set with control name SAMLResponse; multiple SAML assertions MAY be included in the response. A single target description

MUST be included with the control name set to `TARGET`.

The SAML response MUST include the `Recipient` attribute with its value set to https://`<assertion consumer host name and path>`. At least one of the SAML assertions included within the response MUST be an SSO assertion.

The destination site MUST ensure a “single use” policy for SSO assertions communicated by means of this profile.

> **Note:** The implication here is that the destination site will need to save state. A simple implementation might maintain a table of pairs, where each pair consists of the assertion ID and the time at which the entry is to be deleted (where this time is based on the SSO assertion lifetime.). The destination site needs to ensure that there are no duplicate entries. Since SSO assertions containing authentication statements are recommended to have short lifetimes in the web browser context, such a table would be of bounded size.

Confidentiality and message integrity MUST be maintained for the HTTP request in step 3. It is RECOMMENDED that the assertion consumer URL be protected by SSL 3.0 or TLS 1.0 (see Section). Otherwise, the assertions transmitted in step 3 will be available in plain text to any attacker who might then impersonate the assertion subject.

Every subject-based statement in the assertion(s) returned to the destination site MUST contain a `<saml:SubjectConfirmation>` element. The <ConfirmationMethod> element in the <SubjectConfirmation> MUST be set to urn:oasis:names:tc:SAML:1.0:cm:bearer.

3.4.6 Step 4: Responding to the User’s Request for a Resource

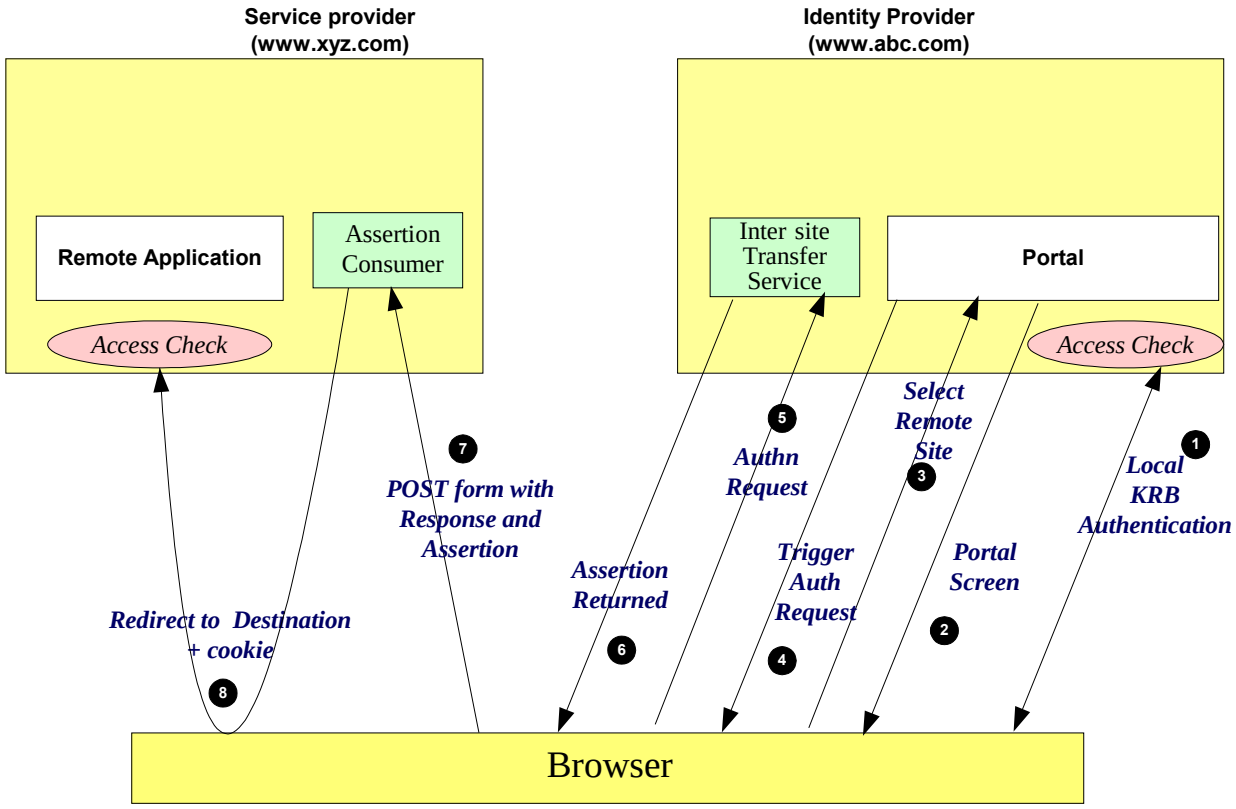
In step 4, the user’s browser is sent an HTTP response that either allows or denies access to the desired resource. The TARGET form element may be used to decide how to respond to the request and what resource to return, possibly via a redirect or some other means,

No normative form is mandated for the HTTP response. The destination site SHOULD provide some form of helpful error message in the case where access to resources at that site is disallowed.

3.4.7 Overall Processing

The following figure shows the processing and message flows for the Browser/POST Kerberos profile.

The workstation contains a Kerberos client with additional SAML functionality. The example describes the use of a Java applet on the workstation to perform part of the processing, however any other suitable technology could be used. It is highly RECOMMENDED that if a Java applet is used then it is loaded locally from the workstation file store.



The processing is as follows:

1. The user authenticates to the source site using Kerberos.
2. At some point the user wishes to use web based applications and may access a "Portal" web page listing resources that they may access.
3. The user selects a menu option (or function) on the Portal web page that means the user wants to access a resource or application on a destination web site www.xyz.com (although, of course, the user may not be made aware of this).
4. The portal application then sends a HTML page to the workstation that causes an applet to be executed. The HTML contains the URL of the resource on the remote site. This is known as the TARGET URL. The HTML also contains binding information for the source site Inter-Site transfer Service. The binding value provided will depend on the binding being used, it will have to uniquely define the location of the ITS on the source site. The HTML page also contains the URL of the destination site's Assertion Consumer service. The HTML page would take the form of:

```
<title>A Kerberos/SAML client</title>
<hr>
<applet code "kerberosSaml.class">
<param name=TARGET value="http://www.xyz.com/index.asp">
<param name=ITS value={binding}>
<param name=CONSUMER value="http://www.xyz.com:8000/AssertionConsumer
</applet>
<hr>
```

5. The workstation application (e.g. the Java applet) sends to the Inter-Site Transfer service an `<AuthnRequest>` using an appropriate binding protocol (as described in the bindings section). The ITS uses the Principal Name from Kerberos to define the Subject.
6. The Inter-site Transfer Service sends a `<SAMLResponse>` back to the Java applet, within which is a SAML assertion. The SAML specifications mandate that the response must be digitally signed.
7. The Java applet will then cause a HTTP POST containing the SAML response to be sent to the

434 destination's (relying party) Assertion Consumer service.

435 8. The replying party's Assertion Consumer validates the digital signature on the SAML Response, if this
436 validates it the sends a redirect to the browser causing it to access the TARGET resource. An access
437 check is then made to establish whether the user has the correct authorization to access the
438 www.xyz.com web site and the TARGET resource. The TARGET resource is the returned to the
439 browser.

4 Bindings

(This section is intended to go into the SAML 2.0 Bindings Document)

4.1 Introduction

This section describes various Kerberos protocol bindings that may be used to pass a Kerberos authenticated identity from a SAML Requester to an Identity Provider, to secure communication between SAML components, or between components and intermediaries. A number of bindings are defined, including:

a. GSSAPI with Kerberos [RFC 1964] bindings;
b. GSSAPI profile of SASL [SASL-GSSAPI] bindings;

Other bindings, not currently defined in SAML 2.0 are discussed in the following section. These bindings, and others MAY be defined in later versions of SAML.

4.2 Browsers, Web Server's and Kerberos

There are many approaches to using Kerberos between a Web Browser and Web Server for the purposes of user authentication, security and Single Sign-On, some of these are summarized below :

4.2.1 Kerberos Client at Web Server

With this approach the authentication of the user is typically performed using basic HTTP authentication, or a Web form based authentication. The Web Server then uses the details entered by the user to perform the authentication and obtain initial Kerberos credentials from the Key Distribution Center (KDC). The credentials are then cached on the Web Server and a domain session cookie is created containing an encrypted copy of the users initial ticket granting ticket (tgt) or a unique value which maps to the cached credentials next time their Browser requests a page.

There are many Kerberos based security solutions available that use, or are based on this approach. Some of the considerations with this approach are:

- **SSL is REQUIRED** - To encrypt the communication between Browser and Web Server since the user's password would be passed across this communication path. This dependency on SSL defeats some of the objectives of using Kerberos as a network security protocol in a Web environment.
- **Flexibility** - Not flexible enough to support all types of user authentication with Kerberos, other than simple username/password. If for example smart card based, or hardware token based authentication is required this approach would be difficult, if not impossible to implement securely with Kerberos.
- **Cookie Dependency** – Cookies are used for session management, so when the user closes their browser and opens it again they would have to re-authenticate. The access to Web resources on multiple domains makes implementation of Single Sign-On difficult, however when SAML is used with Kerberos Single Sign-On with multiple domains is achievable in a standard and interoperable manner.

4.2.2 Kerberos Client at Workstation or Browser

With this approach the Kerberos client implemented on the user's workstation, or in their Browser would be used to authenticate the user and obtain their initial credentials (tgt). These credentials can then be used by non-Web and Web based applications. The Browser can use the user's credentials to authenticate the user to the Web Server. This avoids the need to pass passwords to the Web Server and can therefore be implemented without the dependency on SSL for confidentiality.

A variation on this approach involves a Browser plug-in or Applet that performs authentication of the user either independent of the Kerberos client on the workstation, or using it, and sharing the credential cache to link non-Web and Web-based Single Sign-On to applications.

The following standards are available to authenticate a user at a workstation to a Web Server using Kerberos credentials. In addition, on of the SAML 2.0 bindings described later in this document, SASL [HTTP-SASL] could be used to securely pass the user's identity to the Web Server.

- TLS with Kerberos 5 Cipher Suites as defined in [RFC 2712];

  This RFC proposes the addition of new cipher suites to the TLS protocol to support Kerberos-based authentication. Kerberos credentials are used to achieve mutual authentication and to establish a shared secret which is subsequently used for secure client-server communication.

- SPNEGO / GSS [HTTP-SPNEGO].

  An individual submission Internet draft (expired) - implemented in Microsoft products, e.g. Internet Explorer (IE) and Internet Information Server (IIS). It is also available as a plug-in for many commercial and non-commercial Web Server and Browser's. e.g. Apache has "mod_negotiate", also Mozilla 1.5/1.6 has support for this draft.

4.3 Kerberos Bindings for SAML 2.0

Kerberos can used to provide security between a SAML Requester of an `<AuthnRequest>` and the Identity Provider's `<Response>`. The bindings described below use Kerberos. In each of these bindings the following must be true:

- **Mutual Authentication:** The session between the SAML Requester and Identity Provider must have been authenticated mutually so that two-way electronic trust exists between the two authenticating parties.
- **Message Integrity:** All messages between between the SAML Requester and Identity Provider must be integrity protected to prevent messages exchanged between these parties being tampered with.
- **Confidentiality:** All messages between the SAML Requester and the Identity Provider MAY be encrypted using keys issued by the Kerberos authentication server (KDC). If used, the RECOMMENDED algorithm is AES-128 or AES-256, however DES, 3DES, and RC4 are also ACCEPTABLE.
- **Replay Prevention:** All security context's established using Kerberos should enable replay prevention to avoid replay attack and sequencing errors whilst passing security token's from initiator to acceptor.

The above security features can be implemented in the bindings using parameters passed to the binding protocol API when the security context is initiated.

4.3.1 GSSAPI with Kerberos bindings

Rather than using Kerberos low level API's for ticket requests and credential management for SAML 2.0 the use of GSSAPI [RFC 2743] with a Kerberos mechanism [RFC 1964] is RECOMMENDED. The Kerberos protocol can then be used to :

- Requesting SAML Assertions from an Identity Provider;
- Transferring SAML Assertions across the network in a secure manner;
- Establishing a Security Context between communicating parties in a SAML implementation;
- Signing and Encrypting messages using the security context.

The use of GSSAPI for binding a session in a SAML deployment allows an initiator to use Kerberos, but

531 the acceptor to use an alternative security mechanism (e.g. PKI), or vice-versa. This scenario is
532 described in [IBM-MS-GSS] with examples.
533

534 4.3.2 GSSAPI profile of SASL bindings

535 The SASL GSSAPI [SASL-GSSAPI] exchange carried out over HTTP [HTTP-SASL], or SOAP
536

5 Normalization and SAML Identifiers

(This section is intended to go into a number of the the SAML 2.0 documents)

5.1 Authentication Method Identifiers

5.1.1 Kerberos

URI: urn:ietf:rfc:1510

The authentication was performed by means of the Kerberos protocol [RFC 1510], an instantiation of the Needham-Schroeder symmetric key authentication mechanism [Needham 78]

When the Kerberos protocol is used to authenticate a user a variety of user identification methods are available (depending on the specific Kerberos implementation being used). These methods typically include, username & password, hardware security token, or a smart card containing a user's X.509 certificate. In some cases a combination of these methods are used together (e.g. username, password & security token). After the required information has been used by the Key Distribution Center (KDC) to determine the authenticity of the user a Kerberos ticket granting ticket (tgt) is returned to the initiator of the authentication request.

With SAML 2.0, when the Kerberos protocol is used to authenticate a user the assertion created should include details of the method used to authenticate the user – this will allow the Service Provider to use the additional authentication strength details to make authorization and/or access decisions.

5.2 NameIdentifier Format Identifiers

5.2.1 Kerberos Principal Name

URI: urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos

Indicates that the content of the `<NameIdentifier>` element is in the form of a Kerberos principal name using the format : `name[/instance]@REALM`. The syntax, format and characters allowed for the `name`, `instance` and `REALM` part of a principal name are described in RFC1510 [RFC 1510].

5.3 Kerberos Attributes and Naming

Example of how a Kerberos principal name is carried within a SAML Assertion.

```
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
    MajorVersion="1"
    MinorVersion="1"
    AssertionID="P1YaAztP6UfswxAjax5TPxQ"
    Issuer="www.entegrity.com"
    IssueInstant="2002-06-19T17:05:37.795Z">
    <saml:Conditions NotBefore="2002-06-19T17:00:37.795Z"
                     NotOnOrAfter="2002-06-19T17:10:37.795Z"/>
    <saml:AuthenticationStatement
         AuthenticationMethod="urn:ietf:rfc:1510"
         AuthenticationInstant="2002-06-19T17:05:17.706Z">
         <saml:Subject>
              <saml:NameIdentifier
                   NameQualifier="http://www.cybersafe.ltd.uk/"
                   Format="urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos">
                   talsop@CYBERSAFE.LTD.UK
              </saml:NameIdentifier>
              <saml:SubjectConfirmation>
                   <saml:ConfirmationMethod>
                        urn:oasis:names:tc:SAML:1.0:cm:artifact
```

```
                    </saml:ConfirmationMethod>
                <saml:SubjectConfirmationData>
                        AAGZE1RNQJEFzYNCGAGPjWvtDIRSZ4lWDqBphqA
                    </saml:SubjectConfirmationData>
                </saml:SubjectConfirmation>
            </saml:Subject>
        </saml:AuthenticationStatement>
</saml:Assertion>
```

5.4 Microsoft Windows Kerberos

SAML 2.0 does not define how the Microsoft PAC attributes can be mapped into an Attribute Statement. A later version of SAML MAY define this mapping.

5.5 Distributed Computing Environment (DCE)

The SAML 2.0 Baseline Attributes document describes the format of DCE PAC data in an Assertion.

5.6 Kerberos Service Ticket

A Kerberos Service Ticket (ST) can be carried in a SAML Assertion within the subject confirmation data (see example below) in order to allow the Service Provider (SP) to verify the subject of the Assertion via the Kerberos trust model. This requires that the SP has a Kerberos protocol capability and a copy of the Kerberos key table (created by the Identity Provider) and containing the symmetric key which it needs to decrypt the ST. When the SP has determined the principal name from the ST it can compare this with the principal name from the name given in the `<NameIdentifier>` element.

Another use for passing ST in an Assertion is that authorization data/attributes can be securely carried from one realm to another using Kerberos tickets to secure the communications.

6 References

6.1 Normative References

[RFC 2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, IETF RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt.

[RFC 1510] J. Kohl, C. Neuman. *The Kerberos Network Authentication Requestor (V5).* IETF RFC 1510, September 1993. http://www.ietf.org/rfc/rfc1510.txt.

Please also refer to http://www.ietf.org/internet-drafts/draft-ietf-krb-wg-kerberos-clarifications-05.txt for recent clarification of RFC1510 specification.

[Needham 78] R. Needham et al. *Using Encryption for Authentication in Large Networks of Computers*. Communications of the ACM, Vol. 21 (12), pp. 993-999. December 1978.

[RFC 2712] A Medvinsky, M Hur. *Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)*. IETF RFC 2712, October 1999. http://www.ietf.org/rfc/rfc2712.txt.

[RFC 2222] Myers, J., "Simple Authentication and Security Layer (SASL)", October 1997. http://www.ietf.org/rfc/rfc2222.txt.

Please also refer to : Melnikov, A., "Simple Authentication and Security Layer (SASL)", draft-ietf-sasl-rfc2222bis (work in progress), February 2004. http://www.ietf.org/internet-drafts/draft-ietf-sasl-rfc2222bis-06.txt

[SASL-GSSAPI] Melnikov, A., "SASL GSSAPI mechanisms", draft-ietf-sasl-gssapi (work in progress), November 2003.

[HTTP-SASL] Nystrom, M. and A. Melnikov, "SASL in HTTP/1.1", draft-nystrom-http-sasl (work in progress), February 2004.

[RFC 2743] Generic Security Service Application Program Interface (GSS-API) Version 2, Update 1 (Proposed Standard), January 2000. http://www.ietf.org/rfc/rfc2743.txt

[RFC 1964] The Kerberos Version 5 GSS-API Mechanism (Proposed Standard), June 1996. http://www.ietf.org/rfc/rfc1964.txt

[HTTP-SPNEGO] An expired individual IETF draft for Kerberos authentication over HTTP, October 2002. http://curl.haxx.se/rfc/draft-brezak-spnego-http-04.txt

[IBM-MS-GSS] A document authored by IBM and Microsoft, “Web Services Security Kerberos Bindings, December 2003” located at : http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-security-kerberos.asp or http://www-106.ibm.com/developerworks/library/ws-seckerb/. This document builds on the WS-Security, WS-Trust, and WS-SecureConversation specifications to integrate Kerberos functionality.

?

A. Acknowledgments

The editors would like to acknowledge the contributions of the OASIS Security Services Technical Committee, whose voting members at the time of publication were:

- TBD

B. Revision History

Rev	Date	By Whom	What
01	8th Jan 2004	John Hughes	Initial version.
02	1st Feb 2004	Tim Alsop	Changed format of so a more generic approach is presented with references to complementary bindings and profiles when applicable.
04	15th March 2004	John Hughes, Tim Alsop	Rewrite so that sections of the documents can be easily inserted into the SAML 2.0 normative document set.

7 Notices

OASIS 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. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. 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 implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS 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 implement this specification. Please address the information to the OASIS Executive Director.