Web Single Sign-On Metadata Exchange Protocol

April 2005

Authors

Rajeev Angal, Sun Microsystems
Chris Kaler, Microsoft
Hubert Le Van Gong, Sun Microsystems
Eve Maler, Sun Microsystems
Ari Medvinsky, Microsoft
John Shewchuk, Microsoft

Copyright Notice

No other rights are granted by implication, estoppel or otherwise.

Abstract

This document defines how a service can query an identity provider for metadata that describes the identity-processing protocol suites supported by that provider, to increase the service's ability to communicate successfully and efficiently with the provider.

Status

This specification is an initial public draft release and is provided for review and evaluation only. The authors hope to solicit your contributions and suggestions in the near future. The authors make no warrantees or representations regarding the specifications in any manner whatsoever.

Table of Contents

1. Introduction
2. Notations and Terminology
2.1. Notational Conventions
2.1.1. Normative Outlines
2.2. XML Namespaces
2.3. Compliance
3. Model
4. Acquiring Identity Provider WebSSO Protocol Suites
4.1. Indicating the Identity Provider
4.1.1. HTTP Identity Header
4.1.2. URL Identity Query String Parameter
4.1.3. Default DNS-Based Endpoint Reference
4.1.4. Summary Rules
4.2. Requesting the Identity Provider's Protocol Suite Document
4.2.1. Requesting Supported Suites
4.2.2. Returning Supported Protocol Suites
4.2.3. Protocol Suites
4.2.4. Determining the Protocol Suite to Use
4.2.5. Metadata Acquisition Security
5. Security Considerations
6. Acknowledgements
7. References

1. Introduction

When a client desires identity-based communication with a service, there is a need to establish a common protocol that is supported by both parties. There are several different models which can be employed – specifically the identity provider can support multiple protocols or the target service can support multiple protocols.

When an identity provider supports multiple protocols the target service simply uses its preferred protocol suite to communicate with the identity provider and the identity provider responds correctly.

However, to maximize the set of clients that are supported, a target service may also elect to provide support for multiple protocol suites. This enables the target service to work with identity providers with limited protocol suite support. Moreover, in some cases, the target may need to dynamically determine the protocol suites the identity provider supports.

To address these situations, this document defines a mechanism whereby target services can determine the protocol suites supported by the client's (requestor's) identity provider and use a supported protocol suite for subsequent communication with the identity provider.

That is, to initiate identity-based communication, the target service requires communication with the client's identity provider. However, the identity provider may support different protocol suites, or even different versions of a common protocol suite. This protocol defines a neutral mechanism to determine the supported protocol suites (and versions) thereby enabling the service to determine the right protocol to use to initiate identity processing.

This protocol also defines a standard process for determining the identity provider for a given client (requestor).

2. Notations and Terminology

This section specifies the notations, namespaces, and terminology used in this specification.

2.1. Notational Conventions

The keywords "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 [RFC 2119].

2.1.1. Normative Outlines

This specification uses the following syntax to define normative outlines for messages:

- The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- Characters are appended to elements and attributes to indicate cardinality:
  - "?" (0 or 1)
  - "*" (0 or more)
  - "+" (1 or more)
- The character "|" is used to indicate a choice between alternatives.

- The characters "[" and "]" are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content. Additional children and/or attributes MAY be added at the indicated extension points but MUST NOT contradict the semantics of the parent and/or owner, respectively. If an extension is not recognized it SHOULD be ignored.
- XML namespace prefixes (see [XML-ns]) are used to indicate the namespace of the element being defined.

Additionally, normative text is provided describing elements and attributes, their expected values, and any usage expectations and restrictions. Normative text within this specification takes precedence over normative outlines, which in turn take precedence over any XML Schema and WSDL descriptions that are provided here or referenced from other specifications.

2.2. XML Namespaces

The XML namespace URI that MUST be used by implementations of this specification is:

```
http://schemas.xmlsoap.org/ws/2005/04/ssi
```

where ssi refers to Single Sign-on Interoperability.

The following table lists XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Prefix	XML Namespace	Specification(s)
ssi	http://schemas.xmlsoap.org/ws/2005/04/ssi	This document
s11	http://schemas.xmlsoap.org/soap/envelope/	SOAP 1.1 [SOAP 1.1]
s12	http://www.w3.org/2003/05/soap-envelope	SOAP 1.2 [SOAP 1.2]
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing	WS-Addressing [WS-Addressing]
wsdl	http://schemas.xmlsoap.org/wsdl/	WSDL [WSDL 1.1]
wsp	http://schemas.xmlsoap.org/ws/2004/09/policy	WS-Policy [WS-Policy]
wsx	http://schemas.xmlsoap.org/ws/2004/09/mex	WS-MetadataExchange [WS-Mex]
xs	http://www.w3.org/2001/XMLSchema	XML Schema [Part 1, 2]

2.3. Compliance

A target service or identity provider is not compliant with this protocol if it fails to satisfy one or more of the MUST or REQUIRED level requirements defined herein. A SOAP Node MUST NOT use the XML namespace identifier for this specification (listed in Section 2.2) within SOAP Envelopes unless it is compliant with this specification.

This specification references a number of other specifications (see the table above). In order to comply with this specification, an implementation MUST implement the portions of referenced specifications necessary to comply with the required provisions of this profile. Additionally, the implementation of the portions of the referenced specifications that are specifically cited in this specification MUST comply with the rules for those portions as established in the referenced specification. It is not necessary for

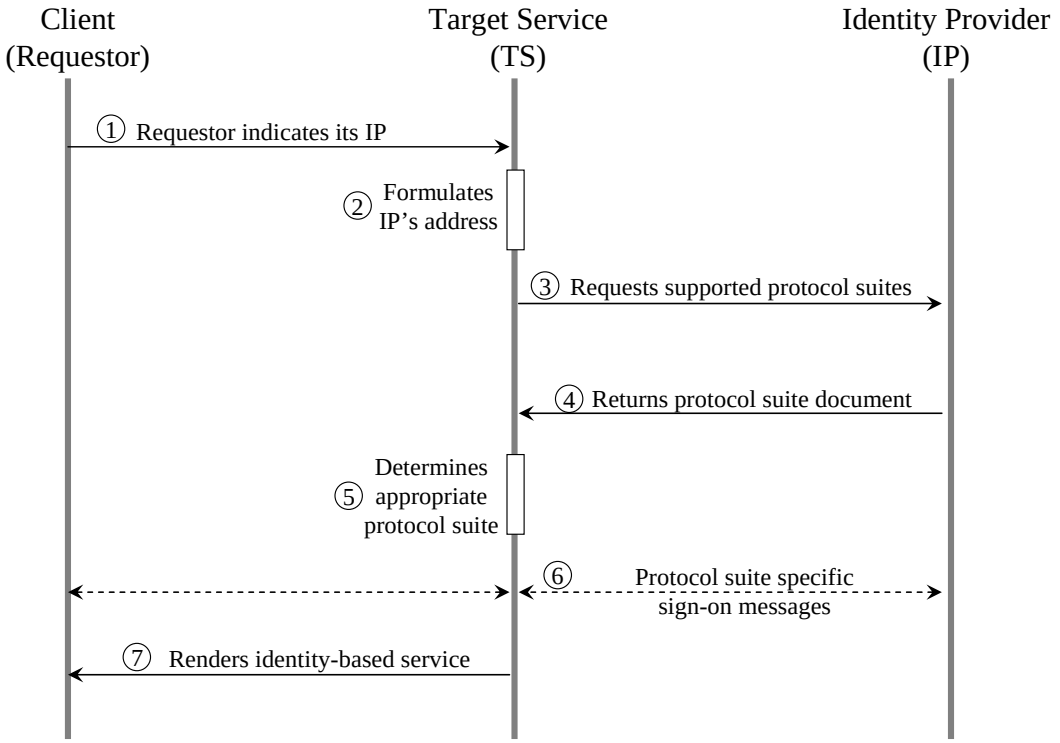
compliance with this specification to implement portions of referenced specifications that are not (directly or transitively) identified by this specification.

Additionally normative text within this specification takes precedence over normative outlines (as described in section 2.1.1), which in turn take precedence over the XML Schema [XML Schema Part 1, Part 2] and WSDL [WSDL 1.1] descriptions. That is, the normative text in this specification further constrains the schemas and/or WSDL that are part of this specification; and this specification contains further constraints on the elements defined in referenced schemas.

3. Model

When a client initiates identity-based communication, the target service initiates a process to determine which identity establishment/verification protocol suite (or version) to use. This protocol suite is used to communicate with the client's identity provider, either directly or indirectly.

The figure below illustrates the general model this profile uses for determining the supported protocol suites.



The key steps are:

1. The client initiates identity-based communication with the target service
2. The target service formulates the address for the identity provider (IP)
3. The target service requests the supported protocol suites from the IP
4. The identity provider returns its supported protocol suites

5. The target service determines the appropriate protocol to use
6. The target service uses a commonly supported protocol suite to communicate with the identity provider and perform single sign-on related operations
7. The target service provides the client with the identity-based service it requested

Step 1

The request indicates, either explicitly or implicitly, the identity provider for the client (requestor). The target service extracts this information.

Step 2

The service needs to request the supported protocol suites from the identity provider. In order to do this, it must first determine where to send requests. For this profile, the target of such requests is an endpoint reference as defined in WS-Addressing.

In some cases an identity provider (or at least its address) is known. In other cases, a user identity reference is known such as *user@ip*. For these cases the URI can be extracted as the right term. It should be noted that if there are privacy concerns around passing identity provider or identifier information, care should be taken to protect this information.

Given the identity provider, an endpoint reference (EPR) can be constructed. This profile defines a specific form allowing specific processing at the identity provider.

Step 3

With an endpoint reference for the identity provider the target service requests the supported protocol suites from the identity provider. This request is done using the WS-MetadataExchange protocol.

This step, and the next step, may not be required if this information is preconfigured or cached.

Step 4

Once an identity provider receives such a request, it returns the relevant document necessary for determining the supported protocol suites for establishing/ verifying identity and any aspects necessary for using that protocol suite. This is done using the WS-MetadataExchange protocol returning a protocol suite document.

Step 5

The target service calculates the intersection of the protocol suites and its own. If choices exist, the target service chooses its preference from among the options.

Step 6

With the protocol suite selected, and options understood, the target service initiates communication with the identity provider either directly or via the requesting client.

Step 7

Once the single sign-on phase is completed, the target service renders the identity-based service the client requested.

It is worth noting that the above model describes the case where there is at least one common protocol suite between the target service and the identity provider. Although highly undesirable, there may be situations where there is no such common protocol suite, in which case the scenario would end at step 5 and what happens after step 5 is beyond the scope of this document.

4. Acquiring Identity Provider WebSSO Protocol Suites

The following sections describe how the identity provider is located and how metadata indicating the provider's supported protocol suites is exchanged and interpreted

4.1. Indicating the Identity Provider

When a request is made, there are a series of steps which are used to determine the identity provider for the requestor:

1. The incoming HTTP request is examined for a special header. If present this header provides information that is used to determine the identity provider. This allows supporting clients to proactively indicate IP information.
2. If the previous step does not yield a result or is unsupported by the target service, then the URL is examined for a special query string parameter that is used to determine the identity provider. This allows portals to augment links to carry IP information.
3. If the previous steps do not yield a result or are unsupported, then custom mechanism may be used to determine identity provider URL or endpoint reference such as a mapping table or a user form prompt. This allows for static agreements to be easily integrated.
4. If step 3 does not yield a result, then an endpoint reference is constructed using the DNS name for the requestor. This provides a fallback mechanism for dynamic or ad-hoc federation scenarios.

4.1.1. HTTP Identity Header

A user agent may provide information indicating how to reach the identity provider. The following HTTP extension header is defined for use with HTTP methods. The header indicates the endpoint reference (EPR) of the requestor's identity provider. The syntax is as follows:

Identity-Provider: EncodedEPR

EncodedEPR: *base64*

The content of this header is an XML-based endpoint reference for the identity provider that is encoded as base64. This encoding is achieved by first encoding the EPR as a UTF-8-based XML 1.0 fragment and then encoding the resultant octet stream as base64. The EPR MUST be an XML conformant fragment. Specifically all namespace declarations must be local to the fragment.

It is RECOMMENDED for interoperability that header lines not exceed 79 characters. For longer base64 encodings, the encoding can be separated and continued. All white space is ignored when parsing the base64 value (each token is concatenated and processed as a single base64 value).

Only a single identity provider value can be specified.

4.1.2. URL Identity Query String Parameter

The following URL query string extension is defined to allow an identity provider URL to be specified as part of a URL request. The syntax is as follows:

URL ? IdentityProvider=*UrlEncodedBase64EncodedEPR*

The value of this parameter is encoded identically to the content of the HTTP Identity Header defined in the previous section except that it additionally applies URL encoding to the base64 encoded value. If this parameter is significant in length, then the sender should consider using an HTTP header due to URL size interoperability concerns.

Only a single identity provider parameter can be specified.

4.1.3. Default DNS-Based Endpoint Reference

If no identity provider is specified, a fallback endpoint reference is constructed using the following pattern:

```
(01) <wsa:EndpointReference
(02)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'>
(03)   <wsa:Address>
(04)       http://DNS-name/IdentityProvider/version/
(05)   </wsa:Address>
(06)   <wsa:ReferenceProperties>
(07)     <ssi:SsiProtocolSuiteHandler xmlns:ssi='...'/>
(08)   </wsa:ReferenceProperties>
(09) </wsa:EndpointReference>
```

Where on line (04) the **DNS-name** is determined from the requestor and the **version** is specified by this document and is defined as **200502-01**. Implementations MAY choose to use a shortened DNS name such as primary domain and suffix, but the choice is implementation specific.

4.1.4. Summary Rules

The following requirements provide a list of mechanisms the service provider MAY use for determining an identity provider, in RECOMMENDED order (from most to least preferred):

R-01) The **Identity-Provider** HTTP header

R-02) The **IdentityProvider** query string parameter

R-03) A custom mapping or user form prompt

R-04) A custom pre-defined pattern based on the requestor's DNS name

4.2. Requesting the Identity Provider's Protocol Suite Document

A service uses the endpoint reference for the identity provider to acquire the protocol suite document from the identity provider.

4.2.1. Requesting Supported Suites

The EPR is applied using the rules defined in WS-Addressing to construct a valid WS-MetadataExchange request to obtain the supported protocol suite document from the identity provider.

As an example, suppose the EPR for the identity provider is as follows:

```
(01) <wsa:EndpointReference
(02)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
```

```
(03)     xmlns:ssi='http://schemas.xmlsoap.org/ws/2005/04/ssi'>
(04)   <wsa:Address>
(05)      http://ipservice.example.org/IdentityProvider
(06)   </wsa:Address>
(07)   <wsa:ReferenceProperties>
(09)     <ssi:SsiProtocolSuiteHandler/>
(10)   </wsa:ReferenceProperties>
(11) </wsa:EndpointReference>
```

The following example illustrates such a request from a service located at http://client.example.com/Endpoint to obtain the supported protocol suites from the identity provider identified by the EPR above.

```
(01) <s12:Envelope
(02)     xmlns:s12='http://www.w3.org/2003/05/soap-envelope'
(03)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
(04)     xmlns:wsx='http://schemas.xmlsoap.org/ws/2004/09/mex'
(05)     xmlns:ssi='http://schemas.xmlsoap.org/ws/2005/04/ssi'>
(06)   <s12:Header>
(07)     <wsa:Action>
(08)       http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request
(09)     </wsa:Action>
(10)     <wsa:MessageID>
(11)         uuid:73d7edfc-5c3c-49b9-ba46-2480caee43e9
(12)     </wsa:MessageID>
(13)     <wsa:ReplyTo>
(14)       <wsa:Address>http://client.example.com/Endpoint</wsa:Address>
(15)     </wsa:ReplyTo>
(16)     <wsa:To>http://ipservice.example.org/IdentityProvider</wsa:To>
(17)     <ssi:SsiProtocolSuiteHandler/>
(18)   </s12:Header>
(19)   <s12:Body>
(20)     <wsx:GetMetadata>
(21)       <wsx:Dialect>
(22)         http://schemas.xmlsoap.org/ws/2005/04/SsiSuites
(23)       </wsx:Dialect>
(24)     </wsx:GetMetadata>
```

```
(25)   </s12:Body>
(26) </s12:Envelope>
```

The following rules are established:

R-05) Protocol policy for identity providers MUST use valid WS-MetadataExchange messages

R-06) Requests for IP protocol policy MUST request a metadata document of dialect http://schemas.xmlsoap.org/ws/2005/04/SsiSuites

R-07) Compliant requests MUST include a <wsa:To> header block as defined in WS-Addressing

R-08) Compliant requests MUST include a <wsa:Action> header block as defined in WS-Addressing using the value defined in WS-MetadataExchange

R-09) Compliant requests MUST include a <wsa:MessageID> header block as defined in WS-Addressing

R-10) Compliant requests MUST include a <wsa:ReplyTo> header block as defined in WS-Addressing

R-11) Compliant requests MUST adhere to WS-Addressing processing rules if an endpoint reference was used to address the message

R-12) Endpoints MUST conform to WS-Addressing syntax and processing rules.

4.2.2. Returning Supported Protocol Suites

The identity establishment/validation protocol suites (and versions) supported are expressed in a protocol suite description document and returned using the response mechanisms defined in WS-MetadataExchange.

The following example illustrates such a response.

```
(01) <s12:Envelope
(02)     xmlns:s12='http://www.w3.org/2003/05/soap-envelope'
(03)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
(04)     xmlns:wsp='http://schemas.xmlsoap.org/ws/2004/09/policy'
(05)     xmlns:wsx='http://schemas.xmlsoap.org/ws/2004/09/mex' >
(06)   <s12:Header>
(07)     <wsa:Action>
(08)       http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response
(09)     </wsa:Action>
(10)     <wsa:MessageID>
(11)         uuid:73d7edfc-5c3c-49b9-ba46-2481caee4177
(12)     </wsa:MessageID>
(13)     <wsa:RelatesTo>
(14)         uuid:73d7edfc-5c3c-49b9-ba46-2480caee43e9
(15)     </wsa:RelatesTo>
(16)     <wsa:To>http://client.example.com/MyEndpoint</wsa:To>
```

```
(17)   </s12:Header>
(18)   <s12:Body>
(19)     <wsx:Metadata>
(20)       <wsx:MetadataSection
(21)         Dialect='http://schemas.xmlsoap.org/ws/2005/04/SsiSuites' >
(22)         <wsp:ExactlyOne>
(23)           ...
(24)         </wsp:ExactlyOne>
(25)       </wsx:MetadataSection>
(26)     </wsx:Metadata>
(27)   </s12:Body>
(28) </s12:Envelope>
```

The following rules are established:

R-13) Responses to request for protocol policy MUST use valid WS-MetadataExchange messages

R-14) Responses for IP protocol policy MUST provide a specific form of dialect http://schemas.xmlsoap.org/ws/2005/04/SsiSuites

R-15) Compliant responses MUST include a <wsa:To> header block as defined in WS-Addressing

R-16) Compliant responses MUST include a <wsa:Action> header block as defined in WS-Addressing using the value defined in WS-MetadataExchange

R-17) Compliant responses MUST include a <wsa:MessageID> header block as defined in WS-Addressing

R-18) Compliant responses MUST include a <wsa:RelatesTo> header block as defined in WS-Addressing which corresponds to the <wsa:MessageID> header block in the corresponding request

R-19) Compliant responses MUST process the endpoint reference specified in the <wsa:ReplyTo> header block of the corresponding request when constructing responses (including any reference properties)

R-20) The protocol suite description document MUST conform to the protocol suite dialect as defined in this specification

4.2.3. Protocol Suites

To simplify processing, the supported protocols and versions should be specified using a self-describing element. That is, any sub-values or attributes present MUST NOT be used in matching (intersecting) the identifiers.

The semantics of the <wsp:ExactlyOne> are that the receiver of the MEX response message MUST choose to use exactly one of the listed protocols for its further interactions with the client. The <wsp:ExactlyOne> element SHOULD contain at least one subelement indicating the supported protocol suites. It MAY contain more than one subelement. This specification provides for extensibility such that additional protocol suites MAY be defined and selected if mutually agreed upon.

The following rules are established:

R-21) Protocol suite identifiers MUST be considered matching if their identifiers (namespace and local part of the element name) match.

R-22) Protocol suite documents SHOULD contain a choice within a single `<wsp:ExactlyOne>` element

R-23) Protocol suite choices SHOULD be specific identifiers which inherently define protocols and options

4.2.4. Determining the Protocol Suite to Use

The target service determines whether any of the items on the returned protocol suite document correspond to any of its supported protocol suites.

If there are multiple options to choose from, the service chooses its preference.

The following rules are established:

R-24) Protocol suites are intersected by matching the communicated protocol suite identifiers with those supported by the target service

R-25) The target service MUST select one protocol suite out of the available matches, if any exist

4.2.5. Metadata Acquisition Security

While not required, it is strongly RECOMMENDED that metadata acquisition be secured. Message security is RECOMMENDED, but other forms of security, such as HTTPS MAY be used.

The following rules are established:

R-26) Metadata acquisition SHOULD be secured

R-27) Message security is RECOMMENDED, but HTTPS or another mechanism MAY be used

5. Security Considerations

It is strongly recommended that the messages exchanged by Web services be secured using WS-Security-based [WS-Security] mechanisms. In order to properly secure a message, the SOAP body and all relevant SOAP header blocks need to be explicitly included in the signature's "signed data". Specifically, any standard messaging header blocks, such as those from WS-Addressing [WS-Addressing], need to be included in the same signature as the SOAP body in order to "bind" them all together.

Additionally, different security mechanisms may be desired depending on the frequency of message transmission. For example, for infrequent messages, public key technologies applied to individual messages, as described above, may be adequate. However, for high-frequency message transmissions, it may be more performant to establish a security context between the endpoints. If a shared secret is used, it is RECOMMENDED that derived keys be used to strengthen the secret.

Requests for metadata that are not available to anonymous parties are strongly RECOMMENDED to require usage of WS-Security so that the requester can be authenticated and authorized to access the indicated metadata. Similarly, integrity and confidentiality SHOULD be used whenever metadata has restricted access.

Recipients of metadata are RECOMMENDED to validate the signature to authenticate and verify the integrity of the data. Specifically, recipients SHOULD verify that the sender has the right to "speak" for the metadata. This is important because some metadata,

such as schemas, have embedded target URIs that might be outside the scope of the sender.

If a metadata request results in a reference to another location, care should be taken if that location is in a different security domain or realm from that of the original request target.

It should be noted that when using URL parameters to indicate the identity providers there is the possibility of a redirect attack by inserting a different identity provider than the requestor expected (because the URL parameters are often not verified by users). Constraints on the identity provider, additional security mechanisms, and/or user interface should be used to mitigate against such attacks

The following list summarizes common classes of attacks that apply to this protocol and identifies the mechanism to prevent/mitigate the attacks:

- **Message alteration** – Alteration can be prevented through including signatures of the message information using WS-Security mechanisms.
- **Message disclosure** – Confidentiality can be preserved by encrypting sensitive data using WS-Security mechanisms.
- **Key integrity** – Key integrity can be maintained by using the strongest algorithms possible.
- **Authentication** – Authentication of messages can be established using the mechanisms described in WS-Security.
- **Accountability** – Accountability is a function of the type of and strength of the key and algorithms being used. In many cases, a strong symmetric key provides sufficient accountability. However, in some environments, strong PKI signatures are required.
- **Availability** – Metadata services are subject to a variety of availability attacks such as application-level denial of service. It is recommended that the mechanisms described in WS-Security be considered as mitigations for some forms of attacks. Other attacks, such as network-level denial of service, are harder to avoid. Note that both of these classes of attack are outside the scope of this specification.
- **Replay** – Messages may be replayed for a variety of reasons. To detect and eliminate this attack, mechanisms should be used to identify replayed messages such as the timestamp/nonce outlined in WS-Security. Alternatively, and optionally, other technologies, such as sequencing, can also be used to prevent replay of application messages.
- **Privacy** - Adequate privacy protections should be assured so as to inhibit the unauthorized disclosure of personally identifiable information. In addition, controls should be established so that personally identifiable information is not shared without user notification and consent and that where applicable privacy regulations may be accommodated.

6. Acknowledgements

This specification has been developed as a result of joint work with many individuals and teams, including:

Qingwen Cheng (Sun)
Gary Ellison (Former co-author)
Jeff Hodges (Former co-author)

Chuck Mortimore
Jeffrey Schlimmer (Microsoft)
Don Schmidt (Microsoft)
Wei Sun (Sun)
Emily Xu (Sun)
Pat Patterson (Sun)

7. References

[WS-MEX]
Keith Ballinger, et al, "Web Services Metadata Exchange (WS-MetadataExchange)", September 2004.

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

[SOAP 1.1]
D. Box, et al, "Simple Object Access Protocol (SOAP) 1.1," May 2000.

[SOAP 1.2]
M. Gudgin, et al, "SOAP Version 1.2 Part 1: Messaging Framework," June 2003.

[WS-Addressing]
D. Box, et al, "Web Services Addressing (WS-Addressing)," August 2004.

[WS-Policy]
S. Bajaj, et al, "Web Services Policy Framework (WS-Policy)," September 2004.

[WS-Security]
A. Nadalin, et al, "Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)," March 2004.

[WSDL 1.1]
E. Christensen, et al, "Web Services Description Language (WSDL) 1.1," March 2001.

[XML Schema, Part 1]
H. Thompson, et al, "XML Schema Part 1: Structures," May 2001.

[XML Schema, Part 2]
P. Biron, et al, "XML Schema Part 2: Datatypes," May 2001.

[XML-ns]
W3C Recommendation, "Namespaces in XML," 14 January 1999