

## 2 **WS-Trust 1.3 Interoperability Profile:** 3 **X.509 Token Profile**

### 4 **Working Draft 01** 5 **07 January 2008**

#### 6 **Specification URIs:**

7 urn:mace:switch.ch:doc:wst:profile:x509-200801

#### 8 **This Version:**

9 <http://switch.ch/grid/support/documents/wst-x509-wd01.pdf>

#### 10 **Previous Version:**

11 No previous version

#### 12 **Latest Version:**

13 <http://switch.ch/grid/support/documents/>

#### 14 **Editors:**

15 Chad La Joie, SWITCH

#### 16 **Related Work:**

17 This document relates to the WS-Trust Interoperability Profile [WST-Interop] and the Web  
18 Services Security: X.509 Token Profile [WSS-X509].

#### 19 **Abstract:**

20 This profile provides the semantics for the use of a Username security token within messages  
21 that comply with the WS-Trust Interoperability Profile.

#### 22 **Status:**

23 Working Draft

24 **Table of Contents**

25 1 Introduction.....3

26 1.1 Notation.....3

27 1.2 Normative References.....3

28 2 X.509 Digital Certificate Token Processing.....5

29 2.1 <wsse:BinarySecurityToken> Usage.....5

30 2.2 X.509 Digital Certificate Token Claim Processing.....5

31 2.3 X.509 Entity Certificate Token Creation.....5

32 2.4 Delegation Support.....5

33 3 Example (informative).....6

34

# 1 Introduction

This profile provides the semantics for the use of an X.509 security token within messages that comply with the WS-Trust Interoperability Profile.

## 1.1 Notation

This specification uses normative text to define an extension to the SAML V2.0 metadata specification.

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC 2119]:

...they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)...

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

Listings of XML schemas appear like this.

Example code listings appear like this.

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:

Prefix	XML Namespace	Comments
ds:	<a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>	This is the XML Signature namespace [XMLSig].
wsse:	<a href="http://docs.oasis-open.org/wss.2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">http://docs.oasis-open.org/wss.2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</a>	This is the WS-Security extension 1.0 namespace defined in [WSS].
wst:	<a href="http://docs.oasis-open.org/ws-sx/ws-trust/200512">http://docs.oasis-open.org/ws-sx/ws-trust/200512</a>	This is the WS-Trust 1.3 namespace defined in [WS-Trust].
wsu:	<a href="http://docs.oasis-open.org/wss.2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">http://docs.oasis-open.org/wss.2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</a>	This is the WS-Security utility namespace defined in [WSS].
xenc:	<a href="http://www.w3.org/2001/04/xmlenc#">http://www.w3.org/2001/04/xmlenc#</a>	This is the XML Encryption namespace defined in .

This specification uses the following typographical conventions in text: <WSTrustElement>, <ns:ForeignElement>, Attribute, **Datatype**, OtherKeyword.

## 1.2 Normative References

- [RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997. See <http://www.ietf.org/rfc/rfc2119.txt>.
- [RFC3820]** S. Tuecke, et al. *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*. IETF RFC 3820, June 2004. See <http://www.ietf.org/rfc/rfc3280.txt>.
- [WSI-BSP]** M. McIntosh, et al. *Basic Security Profile Version 1.0*. Web Services Interoperability Organization, 3 March 2007. See <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>.

65       **[WSS]**               A. Nadalin, et al. *Web Services Security: SOAP Message Security 1.1 (WS-*  
66                           *Security 2004)*. OASIS, 1 February 2004. See <http://www.oasis->  
67                           [open.org/committees/download.php/16790/wss-v1.1-spec-os-](http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf)  
68                           [SOAPMessageSecurity.pdf](http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf).

69       **[WSS-X509]**       A. Nadalin, et al. *Web Services Security: X.509 Certificate Token Profile 1.1.*  
70                           OASIS, 1 February 2006. See <http://www.oasis->  
71                           [open.org/committees/download.php/16785/wss-v1.1-spec-os-](http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf)  
72                           [x509TokenProfile.pdf](http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf).

73       **[WS-Trust]**       A. Nadalin, et al. *WS-Trust 1.3*. OASIS, 19 March 2007. See <http://docs.oasis->  
74                           [open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf](http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf).

75       **[WST-Interop]**   C. La Joie. *WS-Trust Interoperability Profile*. See

76       **[XMLSig]**       D. Eastlake, et al. *XML-Signature Syntax and Processing*, World Wide Web  
77                           Consortium, February 2002. See <http://www.w3.org/TR/xmlsig-core/>.

## 78 2 X.509 Digital Certificate Token Processing

### 79 Token Identification:

80 `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-`  
81 `profile-1.0#X509v3`

### 82 2.1 <wsse:BinarySecurityToken> Usage

83 In order to improve the chance of interoperability when using the WSS: X.509 Certificate Token Profile  
84 1.1 [WSS-X509] defined <wsse:BinarySecurityToken> such tokens MUST also conform to the  
85 WS-I Basic Security Profile 1.0 [WSI-BSP], section 12.

### 86 2.2 X.509 Digital Certificate Token Claim Processing

87 When an X.509 certificate token type is used as the claim within a request the following steps should be  
88 taken to validate the claim:

- 89 ● The token service MUST validate the certificate has not expired.
- 90 ● The recipient MUST validate that the certificate is issued by a trusted authority. It MAY use PKIX  
91 path validation to perform this.
- 92 ● The token service MUST validate that the certificate has not be revoked. It MAY use CRLs,  
93 OCSP, or some other implementation-dependent mechanism to do this.
- 94 ● The recipient MUST validate that the requester is in possession of the certificate's private key  
95 using the signature challenge defined in section 8.

### 96 2.3 X.509 Entity Certificate Token Creation

97 When a token service issues an X.509 entity certificate token it MUST produce a token with the following  
98 criteria:

- 99 ● The certificate MUST be an X.509, version 3, certificate. The certificate MAY contain extensions  
100 as determined by the token service.
- 101 ● If a public key is provided by the requester, through <wst:UseKey>, this key MUST be used as  
102 the public key of the certificate.
- 103 ● If no key is provided the token service MUST generate an RSA key of at least 2048 bits. The  
104 generated key MUST be returned with security token by within the response's  
105 <wst:RequestedProofToken>.

#### 106 2.3.1 Extensions Parameters

107 The following request extensions parameters MAY be used to convey additional information to the token  
108 service. This information MAY be used during the processing of an X.509 token or token request.

109 `/wst:RequestSecurityToken/wst:UseKey`

110 Indicates the public key that a created X.509 token MUST use.

## 111 **2.4 Delegation Support**

112 In order to support the delegation of an X.509 certificate the token service MUST have access to the  
113 private key for the certificate to be delegated. If the public key was provided to the token service by  
114 requester the requester may also provide the private key by using `<wst:Entropy>`. If the key pair was  
115 generated by the token service, and the token service supports delegated tokens, then it MUST retain the  
116 private key. If the token service does not have access to the private key and delegation is request a fault  
117 is generated.

118 The result of an X.509 certificate delegation is an X.509 proxy certificate [RFC3820].

### 119 3 Example (informative)

120 The following is an example of an x.509 security token.

121 Namespace declarations are not shown in the examples for the sake of brevity.

122

123

```
124 <wsse:BinarySecurityToken wsse:Id="X509SecurityToken"  
125     EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-  
126 wss-x509-token-profile-1.0#Base64Binary">
```

```
127 MIICrzCCAhgCAQAwDQYJKoZIhvcNAQEEBQAwZ8xCzAJBgNVBAYTAkNIMUAWPgYDVQQKEzdTV01U  
128 Q0ggLSBUZWxlaW5mb3JtYXRpa2RpZW5zdGUgZnV1ciBMZWhyZSB1bmQgRm9yc2NodW5nMQwwCgYD  
129 VQQLEwNBQUkxIjAgBgNVBAMTGVNXSVRDSGFhaSBNZXRhZGF0YSB0aWduZXIeHDAaBgkqhkiG9w0B  
130 CQEWDWFhaUBzd2l0Y2guY2gwHhcNMDUwODAzMTEyMjUxWjCBnzELMAkG  
131 A1UEBhMCQ0gxQDA+BgNVBAoTN1NXSVRDSGAtIFRlbGVpbnZvcmlhdGlrZGllbnN0ZSBmdWVvIExl  
132 aHJlIHVuZCBGb3JzY2h1bmcxDDAKBgNVBAsTAFBSTEiMCAGAlUEAxMzU1dJVENIYWVpIE1ldGFk  
133 YXRhIFNpZ25lcjEcMBoGCSqGSIb3DQEJARYNYWVpQHN3aXRjaC5jaDCBnzANBgkqhkiG9w0BAQEF  
134 AAOBjQAwgYkCgYEAsmyBYNZ8mKYutdyQShzuOgnVxDPlUBZE+57S2ORZg1qi4JExOJEPnviHuh6H  
135 Ea1j1hAMGHxr656paDpfXkmGq/Ybk3xmXy2FTnFGpjFpZUV6dY/oJ82rve27C/NVcwZw2nYR15C5  
136 aCCGx/Q1WsbTww+9972141+wBDH7dXlJ+UGkCAwEAATANBgkqhkiG9w0BAQQFAAOBgQCCLuNwTINK  
137 fhBlVC IuTixR1R6mYu/+4KUJWtH1RCOUZhSLFept8HxEvfwuX9xm+Q6Ju/sOgmI1INuSstUGWwV  
138 y0AbpCphUDDmIh9A85ye8DrVaBHQrj5b/JEjCvkY0zhLJzgdzZ6btT40TuCnk2GpdAClu5SyCTiy  
139 56+zDYqPgg==  
140 </wsse:BinarySecurityToken>
```