



SAML V2.0 Holder-of-Key Assertion Profile Version 1.0

Committee Draft 03

20 October 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key-cd-03.html>
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key-cd-03.odt> (Authoritative)
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key-cd-03.pdf>

Previous Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key-cs-01.html>
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key-cs-01.odt> (Authoritative)
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key-cs-01.pdf>

Latest Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key.html>
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key.odt>
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key.pdf>

Technical Committee:

OASIS Security Services TC

Chair(s):

Hal Lockhart, BEA Systems, Inc.
Thomas Hardjono, MIT

Editors:

Tom Scavo, National Center for Supercomputing Applications (NCSA)

Contributors:

Nate Klingenstein, Internet2
Scott Cantor, Internet2

Declared XML Namespace(s):

`urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key`

Abstract:

The *SAML V2.0 Holder-of-Key Assertion Profile* describes the issuing and processing of holder-of-key SAML assertions. Specifically, we show how a SAML issuer binds X.509 data to a `<ds:KeyInfo>` element and how a relying party confirms that a `<ds:KeyInfo>` element

35 matches given X.509 data. The binding material used by the SAML issuer and the matching data
36 used by the relying party are obtained from an X.509 certificate.

37 **Status**

38 This document was last revised or approved by the SSTC on the above date. The level of
39 approval is also listed above. Check the current location noted above for possible later revisions
40 of this document. This document is updated periodically on no particular schedule.

41 TC members should send comments on this specification to the TC's email list. Others
42 should send comments to the TC by using the "Send A Comment" button on the TC's
43 web page at <http://www.oasis-open.org/committees/security>.

44 For information on whether any patents have been disclosed that may be essential to
45 implementing this specification, and any offers of patent licensing terms, please refer to the IPR
46 section of the TC web page (<http://www.oasis-open.org/committees/security/ipr.php>).

47 The non-normative errata page for this specification is located at [http://www.oasis-](http://www.oasis-open.org/committees/security)
48 [open.org/committees/security](http://www.oasis-open.org/committees/security).

Notices

50 Copyright © OASIS Open 2008–2009. All Rights Reserved.

51 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
52 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

53 This document and translations of it may be copied and furnished to others, and derivative works that
54 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published,
55 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
56 and this section are included on all such copies and derivative works. However, this document itself may
57 not be modified in any way, including by removing the copyright notice or references to OASIS, except as
58 needed for the purpose of developing any document or deliverable produced by an OASIS Technical
59 Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be
60 followed) or as required to translate it into languages other than English.

61 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
62 or assigns.

63 This document and the information contained herein is provided on an "AS IS" basis and OASIS
64 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
65 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
66 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
67 PARTICULAR PURPOSE.

68 OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
69 necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to
70 notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such
71 patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced
72 this specification.

73 OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any
74 patent claims that would necessarily be infringed by implementations of this specification by a patent
75 holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR
76 Mode of the OASIS Technical Committee that produced this specification. OASIS may include such
77 claims on its website, but disclaims any obligation to do so.

78 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
79 might be claimed to pertain to the implementation or use of the technology described in this document or
80 the extent to which any license under such rights might or might not be available; neither does it represent
81 that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to
82 rights in any document or deliverable produced by an OASIS Technical Committee can be found on the
83 OASIS website. Copies of claims of rights made available for publication and any assurances of licenses
84 to be made available, or the result of an attempt made to obtain a general license or permission for the
85 use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS
86 Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any
87 information or list of intellectual property rights will at any time be complete, or that any claims in such list
88 are, in fact, Essential Claims.

89 The name "OASIS" is a trademark of [OASIS](#), the owner and developer of this specification, and should be
90 used only to refer to the organization and its official outputs. OASIS welcomes reference to, and
91 implementation and use of, specifications, while reserving the right to enforce its marks against
92 misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

93 **Table of Contents**

94 1 Introduction.....5
95 1.1 Notation.....5
96 1.2 Terminology.....5
97 1.3 Normative References.....6
98 1.4 Non-normative References.....6
99 2 SAML V2.0 Holder-of-Key Assertion Profile.....7
100 2.1 Required Information.....7
101 2.2 Profile Description.....7
102 2.3 X.509 Certificate Usage.....7
103 2.4 Issuing Holder-of-Key Assertions.....8
104 2.4.1 KeyInfo Usage.....8
105 2.4.2 Example.....9
106 2.5 Processing Holder-of-Key Assertions.....10
107 2.6 Security and Privacy Considerations.....11
108 2.6.1 ASN.1 Encoding.....11
109 2.6.2 X.509 Serial Number.....12
110 3 Conformance.....13
111 3.0.1 SAML V2.0 Holder-of-Key Assertion Profile.....13
112 Appendix A. Acknowledgments.....14
113 Appendix B. Revision History.....15
114

1 Introduction

The *SAML V2.0 Holder-of-Key Assertion Profile* describes the issuing and processing of a holder-of-key SAML assertion, that is, an assertion containing a `<saml:SubjectConfirmation>` element whose `Method` attribute is set to `urn:oasis:names:tc:SAML:2.0:cm:holder-of-key`. Specifically, we describe the structural characteristics of a `<ds:KeyInfo>` element with bound X.509 data and show how a relying party confirms that such a `<ds:KeyInfo>` element matches given X.509 data. The binding material used by the SAML issuer and the matching data used by the relying party are obtained from an X.509 certificate.

This profile involves a SAML issuer and a SAML relying party, each with an X.509 certificate in its possession. The SAML issuer uses its certificate to produce a holder-of-key SAML assertion. The relying party consumes the assertion, confirming the attesting entity by comparing the X.509 data in the assertion with the X.509 data in its possession.

1.1 Notation

This specification uses normative text. 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 [RFC2119]:

...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
saml:	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace defined in the SAML V2.0 core specification [SAML2Core].
ds:	http://www.w3.org/2000/09/xmldsig#	This is the XML Signature namespace [XMLSig].
xs:	http://www.w3.org/2001/XMLSchema	This is the XML Schema namespace [Schema1].
xsi:	http://www.w3.org/2001/XMLSchema-instance	This is the XML Schema namespace for schema-related markup that appears in XML instances [Schema1].

This specification uses the following typographical conventions in text: `<SAMLelement>`, `<ns:ForeignElement>`, `Attribute`, **Datatype**, `OtherCode`.

1.2 Terminology

In this specification, a *SAML issuer* is a producer of holder-of-key assertions. Similarly, a *relying party* is a consumer of holder-of-key assertions.

147 A *presenter* transmits a holder-of-key assertion to the relying party. An *attesting entity* is a presenter who
148 is able to satisfy the subject confirmation requirements of the holder-of-key assertion.

149 Usually the attesting entity is the subject of the assertion (hence the terms "subject confirmation" and
150 "confirming the subject"). In general, however, the attesting entity may not be the subject, in which case
151 the previous phrases are misnomers. Thus the terms "attestation" and "confirming the attesting entity" are
152 more technically correct than "subject confirmation" and "confirming the subject," respectively. We will
153 use the term "attesting entity" exclusively in this document.

154 1.3 Normative References

- 155 **[RFC2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
156 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>
- 157 **[RFC4514]** K. Zeilenga. *Lightweight Directory Access Protocol (LDAP): String
158 Representation of Distinguished Names*. IETF RFC 4514, June 2006.
159 <http://www.ietf.org/rfc/rfc4514.txt>
- 160 **[RFC5280]** D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. *Internet
161 X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL)
162 Profile*. IETF RFC 5280, May 2008. <http://www.ietf.org/rfc/rfc5280.txt>
- 163 **[SAML2Core]** OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion
164 Markup Language (SAML) V2.0*. March 2005. [http://docs.oasis-open.org/security/
165 saml/v2.0/saml-core-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)
- 166 **[SAML2Prof]** OASIS Standard, *Profiles for the OASIS Security Assertion Markup Language
167 (SAML) V2.0*. March 2005. [http://docs.oasis-open.org/security/saml/v2.0/saml-
profiles-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-
168 profiles-2.0-os.pdf)
- 169 **[Schema1]** H. S. Thompson et al. *XML Schema Part 1: Structures*. World Wide Web
170 Consortium Recommendation, May 2001. See [http://www.w3.org/TR/2001/REC-
xmlschema-1-20010502/](http://www.w3.org/TR/2001/REC-
171 xmlschema-1-20010502/)
- 172 **[XMLSig]** D. Eastlake, J. Reagle, D. Solo, F. Hirsch, T. Roessler. *XML Signature Syntax
173 and Processing (Second Edition)*. World Wide Web Consortium
174 Recommendation, 10 June 2008. <http://www.w3.org/TR/xmlsig-core/>

175 1.4 Non-normative References

- 176 **[RFC3820]** S. Tuecke, V. Welch, D. Engert, L. Pearlman, M. Thompson. *Internet X.509
177 Public Key Infrastructure (PKI) Proxy Certificate Profile*. IETF RFC 3820, June
178 2004. <http://www.ietf.org/rfc/rfc3820.txt>
- 179 **[RFC4346]** T. Dierks, E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.1*.
180 IETF RFC 4346, April 2006. <http://www.ietf.org/rfc/rfc4346.txt>
- 181 **[SAML2ConDel]** S. Cantor. *SAML V2.0 Condition for Delegation Restriction*. OASIS SSTC
182 Committee Draft 01, 10 March 2009. [http://docs.oasis-
open.org/security/saml/Post2.0/sstc-saml-delegation-cd-01.pdf](http://docs.oasis-
183 open.org/security/saml/Post2.0/sstc-saml-delegation-cd-01.pdf)

2 SAML V2.0 Holder-of-Key Assertion Profile

2.1 Required Information

Identification: urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key

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

SAML Confirmation Method Identifiers: The SAML V2.0 holder-of-key confirmation method identifier (urn:oasis:names:tc:SAML:2.0:cm:holder-of-key) is associated with every <saml:SubjectConfirmation> element issued under this profile.

Description: Given below.

Updates: Supplements the holder-of-key confirmation method described in section 3.1 of [SAML2Prof].

2.2 Profile Description

This specification profiles a type of assertion called a holder-of-key assertion. By definition, a *holder-of-key SAML assertion* contains a <saml:SubjectConfirmation> element whose Method attribute is set to urn:oasis:names:tc:SAML:2.0:cm:holder-of-key. This specification describes how the SAML issuer binds selected X.509 data from an X.509 certificate to the <saml:SubjectConfirmation> element of a holder-of-key assertion. The complementary process involves a relying party who confirms that the X.509 data bound to the assertion matches the data in a given X.509 certificate.

Suppose a SAML response issued by a SAML issuer contains one or more holder-of-key assertions (otherwise this specification is not applicable). At the time the assertion is issued, the issuer possesses an X.509 certificate known to be associated with the attesting entity (who may or may not be present when the assertion is issued). The SAML issuer binds some (or all) of the X.509 data in the certificate to the holder-of-key assertion.

Subsequently, the attesting entity presents the holder-of-key assertion and an X.509 certificate to the relying party. The attesting entity proves possession of the private key corresponding to the public key bound to the certificate, the details of which are out of scope with respect to this profile. The relying party compares the X.509 data in the certificate to the X.509 data bound to the assertion, thereby confirming the attesting entity.

Precisely how the issuer comes to possess a certificate known to be associated with attesting entity and how the assertion and the certificate are presented to the relying party are all out of scope with respect to this profile. On the other hand, the issuing of the holder-of-key assertion itself and the ultimate confirmation of the attesting entity are in scope.

We assume that the relying party trusts the SAML issuer to issue holder-of-key assertions. The SAML issuer, on the other hand, may not even know the intended relying party, so there is no underlying assumption that the SAML issuer trusts the relying party.

2.3 X.509 Certificate Usage

There are no explicit requirements with respect to the X.509 certificate(s) possessed by the SAML issuer and the relying party. If, however, a certificate contains a Subject Key Identifier (SKI) extension, then the certificate MUST be an X.509 v3 certificate [RFC5280]. Other than that, the specific characteristics of these certificates are wholly out of scope with respect to this specification. In particular, there is no expectation that either the SAML issuer or the relying party trusts the issuer of the certificate, and

224 therefore all portions of the certificate, apart from the X.509 data specified in the following sections, are
225 unspecified.

226 The only exception to the above rule is the case where the `<ds:X509Data>` element specified in
227 section 2.4.1 contains a `<ds:X509SubjectName>` element or a `<ds:X509SerialIssuer>` element. In
228 these two cases, the relying party MUST trust the X.509 issuer in order to confirm the attesting entity. This
229 is discussed more fully in section 2.5 below.

230 2.4 Issuing Holder-of-Key Assertions

231 Every assertion containing a holder-of-key `<saml:SubjectConfirmation>` element MUST conform to
232 [SAML2Core] (see section 2.4.1 of Core, especially section 2.4.1.3) and section 3.1 of [SAML2Prof].
233 Where this specification conflicts with the SAML V2.0 specification, the former takes precedence.

234 Suppose a SAML issuer wishes to issue a response containing one or more holder-of-key assertions. As
235 a prerequisite, the SAML issuer MUST possess an X.509 certificate known to be associated with the
236 attesting entity. The SAML issuer binds some or all of the X.509 data in the certificate to the
237 `<saml:SubjectConfirmation>` element of a SAML assertion.

238 Briefly, the SAML issuer binds a `<ds:KeyInfo>` element to the `<saml:SubjectConfirmationData>`
239 element of a holder-of-key assertion. The `<ds:KeyInfo>` element contains one or more of the following
240 elements: `<ds:X509Certificate>`, `<ds:X509SKI>`, `<ds:X509SubjectName>`, or
241 `<ds:X509IssuerSerial>`. A `<ds:X509Certificate>` element contains a base64 encoding of the
242 certificate possessed by the SAML issuer. A `<ds:X509SKI>` element contains the base64 encoding of
243 the Subject Key Identifier (SKI) extension (if there is one) bound to the certificate. A
244 `<ds:X509SubjectName>` element contains the subject distinguished name (DN) bound to the certificate.
245 A `<ds:X509IssuerSerial>` element contains the issuer DN and the issuer serial number bound to the
246 certificate. In each case, the content of the `<ds:KeyInfo>` element conforms to the XML Signature
247 specification [XMLSig]. These requirements are spelled out more clearly in the next section.

248 If the SAML issuer has reason to believe that the relying party trusts the certificate issuer, the SAML issuer
249 MAY include `NotBefore` or `NotOnOrAfter` XML attributes on the
250 `<saml:SubjectConfirmationData>` element. If so, the values bound to the assertion MUST be
251 consistent with the values in the certificate. In particular, the value of the `NotBefore` attribute (resp., the
252 `NotOnOrAfter` attribute) MUST be greater than or equal to (resp., less than or equal to) the `NotBefore`
253 field (resp., the `NotOnOrAfter` field) of the certificate.

254 The `<saml:SubjectConfirmation>` element MAY contain a `<saml:NameID>` element. If it does, the
255 latter identifies an attesting entity different from the subject of the assertion. If the
256 `<saml:SubjectConfirmation>` element does not contain a `<saml:NameID>` element, then the
257 attesting entity and the subject are one and the same.

258 If the `<saml:SubjectConfirmation>` element contains a `<saml:NameID>` element, the attesting
259 entity is presumably acting on behalf of the subject. To more strongly signal such a delegation scenario, a
260 `<saml:Condition>` element MAY be used (cf. [SAML2ConDel]).

261 2.4.1 KeyInfo Usage

262 According to the SAML V2.0 specification, a holder-of-key assertion MUST contain at least one
263 `<ds:KeyInfo>` element within the `<saml:SubjectConfirmationData>` element and that the
264 `<ds:KeyInfo>` element MUST conform to the XML Signature specification. This SAML V2.0 Holder-of-
265 Key Assertion Profile requires that the `<ds:KeyInfo>` element MUST conform to the *Second Edition* of
266 the XML Signature specification [XMLSig] and further constrains the content of each `<ds:KeyInfo>`
267 element to contain exactly one `<ds:X509Data>` element. The `<ds:X509Data>` element MUST NOT
268 contain a `<ds:X509CRL>` element. Instead, the following content options are specified, at least one of
269 which MUST be satisfied:

- 270 • The `<ds:X509Data>` element MAY contain a `<ds:X509Certificate>` element. If it does, the
271 `<ds:X509Certificate>` element MUST contain a base64 encoding of the X.509 certificate
272 possessed by the SAML issuer.
- 273 • The `<ds:X509Data>` element MAY contain a `<ds:X509SKI>` element. If it does, the
274 `<ds:X509SKI>` element MUST contain the base64 encoding of the plain (i.e., *not* DER-encoded)
275 value of the Subject Key Identifier (SKI) extension (as specified in [XMLSig]) of the X.509
276 certificate possessed by the SAML issuer. If the certificate does not contain an SKI extension, the
277 `<ds:X509Data>` element MUST NOT contain a `<ds:X509SKI>` element.
- 278 • The `<ds:X509Data>` element MAY contain a `<ds:X509SubjectName>` element. If it does, the
279 `<ds:X509SubjectName>` element MUST contain the subject distinguished name (DN) bound to
280 the X.509 certificate possessed by the SAML issuer.
- 281 • The `<ds:X509Data>` element MAY contain a `<ds:X509IssuerSerial>` element. If it does,
282 the `<ds:X509IssuerSerial>` element MUST contain the issuer DN and the issuer serial
283 number (as specified in [XMLSig]) bound to the X.509 certificate possessed by the SAML issuer.

284 Use of the `<ds:X509Certificate>` element or the `<ds:X509IssuerSerial>` element is most
285 restrictive since each implies that the exact same certificate is used by both the SAML issuer and the
286 relying party. Use of the `<ds:X509SKI>` element or the `<ds:X509SubjectName>` element is less
287 restrictive since each permits a different certificate to be used by the relying party provided the certificate
288 contains the same key or DN (resp.) as in the certificate used by the SAML issuer.

289 Use of the `<ds:X509SubjectName>` element or the `<ds:X509IssuerSerial>` element is warranted
290 in those situations where the relying party trusts the issuer of the X.509 certificate. The SAML issuer
291 SHOULD NOT bind either of these elements to the `<ds:X509Data>` element unless it knows that such a
292 trust relationship exists.

293 Note that the format of the DN contained in the `<ds:X509SubjectName>` element or the
294 `<ds:X509IssuerSerial>` element is specified in [XMLSig]. In accordance with that specification, it is
295 RECOMMENDED that the DN conform to [RFC4514] in all cases.

296 Since the `<ds:KeyInfo>` element is extensible [XMLSig], other fields or extensions from the X.509
297 certificate may be bound to the holder-of-key assertion. These are provided as a convenience to the
298 relying party, so that the relying party need not have to decode and parse the certificate. All such
299 extensions are out of scope with respect to this profile, however.

300 2.4.2 Example

301 Here is an example of a holder-of-key `<saml:SubjectConfirmation>` element illustrating three of the
302 content options specified in section 2.4:

```
303 <saml:SubjectConfirmation
304   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
305   Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
306   <saml:SubjectConfirmationData
307     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
308     xsi:type="saml:KeyInfoConfirmationDataType">
309     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
310       <ds:X509Data>
311
312         <!-- a base64 encoding of an X.509 certificate -->
313         <ds:X509Certificate>
314 MIIDuDCCAqACCQCJZK8wF0xVXjANBgkqhkiG9w0BAQQFADCBnTELMAkGA1UEBhMCQlIxZzARBgNV
315 BAgtClNvbWUtU3RhdGUxEjAQBgNVBACTCVNBvWUtQ2l0eTESMBAGA1UEChMJR1NvYyAyMDA4MREw
316 EAYDVQQLEwlHU29DIDIwMDgxZzAVBGNVBAmtDkpvYW5hIFRyaW5kYW5kYWRlMSQwIgYJKoZIhvcNAQkB
317 FhVzb211LWFkZHZHJlc3NAAG9zdC5vcmcwHhcNMDgwNjE2MTcyMTQzWhcNMDkwNjE2MTcyMTQzWjCB
318 nTELMAkGA1UEBhMCQlIxZzARBgNVBAGTCVNBvWUtU3RhdGUxEjAQBgNVBACTCVNBvWUtQ2l0eTES
319 MBAGA1UEChMJR1NvYyAyMDA4MREwEAYDVQQLEwlHU29DIDIwMDgxZzAVBGNVBAmtDkpvYW5hIFRy
320 aW5kYWRlMSQwIgYJKoZIhvcNAQkBFhVzb211LWFkZHZHJlc3NAAG9zdC5vcmcwggEiMA0GCSqGSIb3
```

```

321 DQEBAQUAA4IBDwAwggEKAoIBAQDIDVKdO2CCVYA0TspOPmcSNnivjQq7jCacrgRPawKi3/pTuvnW
322 3c2XCpyT2s6Sks3Eg5T4HIXta5E+lOpN8VbTunVdSrac54r2uK8x+8AqX7M0wQw+98iGw9E2an5q
323 xRZfqqE1T5jWL/a/G1/e2TG1mp521W3k1nNtf8rYH39JpwBSZMeW7uHOSZOkT/pVvqPTgG7vUQT6
324 BiRh7PfwslrL0MubbeQ6Z2m3Vnsv20E1FbPzswszh4X1gXj9bnyI2UsuoisW9Y4p4byjL3GJ/hxp
325 mjRjXs+aIpzi0V3MH+jVJ98eomhlUFLaE83xycC8lns+FcCSQZ8RsbnaLZrtC8r7AgMBAAEwDQYJ
326 KoZlIhvcNAQEEBQADggEBACwnWSEpwq5aE7QBdDNNXyok34RIonYi9690yw7i+JU7R/QdE42GERJS
327 DVKBN959ELLJf5d0vybGv08QWbZVQ7eBGn9xaZ7MhSnb1YNDXs9vuv1V2Dy32q1J5nCSzqpJDyln
328 lVFWe9UQMCJOO6ibUtWLhiDQ49kmMabgyYfx28qB6oRdVL+mDI/XTt+mkCgk4Rs78n4kbX6qnRlj
329 dE/YnibP1A7iMh8pQkv49J6sP9SeUmQ2zxKct3tSRzzyypWc8JjOZGuBYGQH19Xm7WEs4CXs7iZJW
330 E32frMatavMcTM/gnDtCc8tZAx12PSLOF1954vapfMjBhg3VTI6QRW//wPE=
331 </ds:X509Certificate>
332
333 <!-- the above X.509 certificate does not contain a
334 Subject Key Identifier extension so the SAML issuer
335 must not include a <ds:X509SKI> element -->
336
337 <!-- the subject DN (in RFC 5414 format) bound to the
338 above X.509 certificate -->
339 <ds:X509SubjectName>emailAddress=some-address@host.org,CN=Joana
340 Trindade,OU=GSoc 2008,O=GSoc 2008,L=Some-City,ST=Some-
341 State,C=BR</ds:X509SubjectName>
342
343 <!-- the issuer DN (in RFC 5414 format) and the issuer serial
344 number (in decimal) bound to the above X.509 certificate -->
345 <ds:X509IssuerSerial>
346 <ds:X509IssuerName>emailAddress=some-address@host.org,CN=Joana
347 Trindade,OU=GSoc 2008,O=GSoc 2008,L=Some-City,ST=Some-
348 State,C=BR</ds:X509IssuerName>
349 <ds:X509SerialNumber>9900230501951362398</ds:X509SerialNumber>
350 </ds:X509IssuerSerial>
351
352 </ds:X509Data>
353 </ds:KeyInfo>
354 </saml:SubjectConfirmationData>
355 </saml:SubjectConfirmation>

```

356 A relying party can confirm the attesting entity by the matching the available X.509 data to any of the
357 above child elements of the <ds:X509Data> element.

358 2.5 Processing Holder-of-Key Assertions

359 The attesting entity presents a holder-of-key assertion and an X.509 certificate to the relying party. The
360 attesting entity **MUST** prove possession of the private key corresponding to the public key bound to the
361 certificate, the details of which are out of scope with respect to this profile. The relying party confirms the
362 attesting entity by comparing the X.509 data in the certificate to the X.509 data bound to the assertion. If
363 the X.509 data in the certificate matches the X.509 data bound to the assertion, the attesting entity is said
364 to be *confirmed*.

365 Regardless of the protocol used, any assertions relied upon **MUST** be valid according to the processing
366 rules specified in [SAML2Core]. In particular, the relying party **MUST** verify the signature (if any) on each
367 assertion containing a holder-of-key <saml:SubjectConfirmation> element. Any assertion that is not
368 valid, or whose subject confirmation requirements cannot be met, **SHOULD** be discarded and **SHOULD**
369 **NOT** be used to establish a security context for the subject.

370 If the <ds:X509Data> element contains multiple child elements, the relying party may choose to confirm
371 the attesting entity based on any one of them. Specifically, the relying party **MUST** confirm that the
372 certificate matches the content of the <ds:X509Data> element as follows:

- 373 • If the <ds:X509Data> element contains a <ds:X509Certificate> element, and the relying
374 party chooses to confirm the attesting entity based on this element, the relying party **MUST** ensure
375 that the certificate bound to the assertion matches the X.509 certificate in its possession.

376 Matching is done by comparing the base64-decoded certificates, or the hash values of the
377 base64-decoded certificates, byte-for-byte.

- 378 • If the `<ds:X509Data>` element contains a `<ds:X509SKI>` element, and the relying party
379 chooses to confirm the attesting entity based on this element, the relying party MUST ensure that
380 the value bound to the assertion matches the Subject Key Identifier (SKI) extension bound to the
381 X.509 certificate. Matching is done by comparing the base64-decoded SKI values byte-for-byte.
382 If the X.509 certificate does not contain an SKI extension, the attesting entity is not confirmed and
383 the relying party SHOULD disregard the assertion.
- 384 • If the `<ds:X509Data>` element contains a `<ds:X509SubjectName>` element, and the relying
385 party chooses to confirm the attesting entity based on this element, the relying party MUST ensure
386 that the subject distinguished name (DN) bound to the assertion matches the DN bound to the
387 X.509 certificate. If, however, the relying party does not trust the certificate issuer to issue such a
388 DN, the attesting entity is not confirmed and the relying party SHOULD disregard the assertion.
- 389 • If the `<ds:X509Data>` element contains a `<ds:X509IssuerSerial>` element, and the relying
390 party chooses to confirm the attesting entity based on this element, the relying party MUST ensure
391 that the issuer DN and issuer serial number bound to the assertion match the issuer DN and the
392 issuer serial number (resp.) bound to the X.509 certificate. If the relying party does not trust the
393 certificate issuer to issue X.509 certificates, however, the attesting entity is not confirmed and the
394 relying party SHOULD disregard the assertion.

395 In the case of a `<ds:X509Certificate>` element or a `<ds:X509SKI>` element, the matching process
396 is relatively straightforward. If the `<ds:X509Data>` element contains a `<ds:X509SubjectName>`
397 element or a `<ds:X509IssuerSerial>` element, however, and the relying party chooses to confirm the
398 attesting entity based on one of these elements, the relying party MUST trust the issuer of the X.509
399 certificate before the attesting entity can be considered confirmed. If such a trust relationship between the
400 relying party and the certificate issuer does not exist, the relying party SHOULD disregard the assertion.

401 If the `<saml:SubjectConfirmationData>` element includes `NotBefore` or `NotOnOrAfter`
402 attributes, and the relying party trusts the issuer of the X.509 certificate, the relying party MUST confirm
403 that the current time is greater than or equal to (resp., less than or equal to) the value of the `NotBefore`
404 (resp., the `NotOnOrAfter`) attribute. If this requirement is not met, the attesting entity is not confirmed
405 and the relying party SHOULD disregard the assertion.

406 2.6 Security and Privacy Considerations

407 This profile assumes that both the SAML issuer and the relying party have access to an X.509 certificate.
408 For those deployments that wish to avoid or do not require an X.509-based public key infrastructure (PKI),
409 this may seem unnecessarily restrictive. In fact, the use of X.509 certificates is typical and provides a
410 number of advantages. First, observe that the SSL/TLS protocol [RFC4346] requires the use of X.509
411 certificates. Second, and most importantly, since there is no presumption of an underlying trust model for
412 X.509 certificates, the full range of possible content for the `<ds:KeyInfo>` element is avoided. Those
413 deployments that are in fact based on such a trust model, or wish to avoid X.509 certificates altogether,
414 may choose to profile additional child elements such as `<ds:KeyName>` or `<ds:KeyValue>`.

415 Deployments that rely on holder-of-key SAML assertions will no doubt impose their own requirements on
416 the X.509 certificates used to obtain those assertions. For example, some deployments will require the
417 certificate to be an X.509 end-entity certificate [RFC5280] issued by a trusted X.509 certification authority
418 (CA) or a certificate based on a trusted X.509 end-entity certificate (such as an X.509 proxy certificate
419 [RFC3820]). This specification imposes no such restrictions, however.

420 2.6.1 ASN.1 Encoding

421 For compatibility with the XML Signature specification [XMLSig], this profile intentionally avoids any
422 discussion of the ASN.1 encoding of the X.509 certificate possessed by the SAML issuer and the relying

423 party. Indeed, in the case of the `<ds:X509Certificate>` element, the ASN.1 encoding of the
424 certificate doesn't matter. In this case, the SAML issuer simply base64-encodes the ASN.1-encoded
425 certificate in its possession and binds it to the `<ds:X509Certificate>` element. Later the relying party
426 base64-decodes the content of the `<ds:X509Certificate>` element and compares the resulting
427 certificate (byte-for-byte) with the ASN.1-encoded certificate in its possession.

428 In the case of the `<ds:X509SKI>`, `<ds:X509SubjectName>`, or `<ds:X509IssuerSerial>` elements,
429 however, the ASN.1 encoding of the certificates *does* matter. To produce these elements, the SAML
430 issuer must ASN.1-decode the certificate in its possession and parse the ASN.1 to obtain the X.509 data
431 to be bound to the assertion. Likewise the relying party must ASN.1-decode the certificate in its
432 possession, parsing the ASN.1 to obtain the required X.509 data, which it compares to the X.509 data
433 bound to the assertion.

434 The basic problem is that the ASN.1 encoding of an X.509 certificate is not guaranteed. While it is true
435 that an X.509 certificate is often DER-encoded, a robust implementation must be prepared to handle other
436 ASN.1 encodings besides DER, mainly BER and CER. Consequently it is anticipated that deployments
437 will prefer the `<ds:X509Certificate>` element for maximum interoperability. In fact, this preference is
438 reflected in the conformance requirements of this profile (section 3).

439 **2.6.2 X.509 Serial Number**

440 Note that some CAs use large random numbers as serial numbers to prevent sequence guessing.
441 However, not all XML libraries are capable of dealing with large integers in the
442 `<ds:X509IssuerSerial>` element. The problem is that the `<ds:X509SerialNumber>` child element
443 of the `<ds:X509IssuerSerial>` element is typed as an arbitrary integer in [XMLSig] yet conforming
444 implementations are required to support only 18 decimal digits. Thus the `<ds:X509IssuerSerial>`
445 element should be used with care.

446 **3 Conformance**

447 **3.0.1 SAML V2.0 Holder-of-Key Assertion Profile**

448 Both the SAML issuer and the relying party MUST conform to section 2.3.

449 A SAML issuer MUST follow the issuing rules in section 2.4. In particular, a SAML issuer MUST produce
450 `<ds:KeyInfo>` elements that conform to section 2.4.1. Likewise, a relying party MUST follow the
451 processing rules in section 2.5.

452 To claim conformance to this specification, a SAML issuer implementation MUST support the
453 `<ds:X509Certificate>` element specified in section 2.4.1. Support for the remaining child elements
454 specified in section 2.4.1 is OPTIONAL for SAML issuers.

455 Likewise a conforming relying party implementation MUST support the `<ds:X509Certificate>`
456 element specified in section 2.5. Support for the remaining child elements specified in section 2.5 is
457 OPTIONAL for relying parties.

458 **Appendix A. Acknowledgments**

459 The editor would like to acknowledge the contributions of the OASIS Security Services (SAML) Technical
460 Committee, whose voting members at the time of publication were:

- 461 • John Bradley, Individual
- 462 • Scott Cantor, Internet2
- 463 • Duane DeCouteau, Veterans Health Administration
- 464 • Christian Guenther, Nokia Siemens Networks GmbH & Co.
- 465 • Thomas Hardjono, M.I.T.
- 466 • Frederick Hirsch, Nokia Corporation
- 467 • Ari Kermaier, Oracle Corporation
- 468 • Nathan Klingenstein, Internet2
- 469 • Hal Lockhart, Oracle Corporation
- 470 • Paul Madsen, NTT Corporation
- 471 • Kyle Meadors, Drummond Group Inc.
- 472 • Bob Morgan, Internet2
- 473 • Thinh Nguyenphu, Nokia Siemens Networks GmbH & Co.
- 474 • Rob Philpott, EMC Corporation
- 475 • Anil Saldhana, Red Hat
- 476 • Tom Scavo, National Center for Supercomputing Applications
- 477 • Kent Spaulding, Skyworth TTG Holdings Limited
- 478 • David Staggs, Veterans Health Administration
- 479 • Emily Xu, Sun Microsystems

480 The editor would also like to acknowledge the following contributors:

- 481 • Joana M. F. da Trindade, Universidade Federal do Rio Grande do Sul (Brazil)
- 482 • The members of the IETF PKIX Working Group
- 483 • Peter Sylvester, EdelWeb (France)
- 484 • Brett Beaumont, SSC, New Zealand Government

Appendix B. Revision History

Document ID	Date	Committer	Comment
sstc-saml2-holder-of-key-draft-01	7 Aug 2008	T. Scavo	Initial draft
sstc-saml2-holder-of-key-draft-02	14 Aug 2008	T. Scavo	Remove all refs to <code>samlp:</code>
sstc-saml2-holder-of-key-draft-03	7 Sep 2008	T. Scavo	Remove proof of possession requirement
sstc-saml2-holder-of-key-draft-04	6 Oct 2008	T. Scavo	Response to comments
sstc-saml2-holder-of-key-draft-05	20 Oct 2008	T. Scavo	Updated KeyInfo Usage rules
sstc-saml2-holder-of-key-draft-06	13 Nov 2008	T. Scavo	Dropped DER-encoding requirement
sstc-saml2-holder-of-key-draft-07	7 Dec 2008	T. Scavo	Added NotBefore/NotOnOrAfter attributes
sstc-saml2-holder-of-key-draft-08	11 Jan 2009	T. Scavo	Relaxed the X.509 v3 requirement
sstc-saml2-holder-of-key-draft-09	20 Jan 2009	T. Scavo	Response to comments
sstc-saml2-holder-of-key-cd-01	9 Mar 2009	T. Scavo	Committee Draft 01
sstc-saml2-holder-of-key-draft-10	14 Jun 2009	T. Scavo	Response to Public Comments
sstc-saml2-holder-of-key-cd-02	5 Jul 2009	T. Scavo	Committee Draft 02
sstc-saml2-holder-of-key-cs-01	29 Jul 2009	tc-admin	Committee Specification 01
sstc-saml2-holder-of-key-draft-11	4 Oct 2009	T. Scavo	Fixed minor bugs in CS 01
sstc-saml2-holder-of-key-cd-03	3 Nov 2009	T. Scavo	Committee Draft 03