



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

Working Draft 0706, ~~137 Nov~~December 2008

## Specification URIs:

TBD

## Technical Committee:

OASIS Security Services TC

## Chair(s):

Hal Lockhart, BEA Systems, Inc.

Brian Campbell, Ping Identity Corporation

## Editors:

Tom Scavo, National Center for Supercomputing Applications (NCSA)

## Contributors:

Nate Klingenstein, Internet2

Scott Cantor, Internet2

## 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 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 a standard X.509 v3 certificate.

## Status

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

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

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 IPR section of the TC web page (<http://www.oasis-open.org/committees/security/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/security>.

## 34 Notices

35 Copyright © OASIS Open 2008. All Rights Reserved.

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

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

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

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

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

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

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

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

## 78 Table of Contents

79	1 Introduction.....	4
80	1.1 Notation.....	4
81	1.2 Normative References.....	5
82	1.3 Non-normative References.....	5
83	1.4 Conformance.....	5
84	1.4.1 SAML V2.0 Holder-of-Key Assertion Profile.....	5
85	2 SAML V2.0 Holder-of-Key Assertion Profile.....	6
86	2.1 Required Information.....	6
87	2.2 Profile Description.....	6
88	2.3 X.509 Certificate Usage.....	6
89	2.4 Issuing Holder-of-Key Assertions.....	7
90	2.4.1 KeyInfo Usage.....	7
91	2.4.2 Example.....	8
92	2.5 Processing Holder-of-Key Assertions.....	9
93	2.6 Security and Privacy Considerations.....	10
94	2.6.1 ASN.1 Encoding.....	10
95	2.6.2 X.509 Serial Number.....	11
96	Appendix A. Acknowledgments.....	12
97	Appendix B. Revision History.....	13
98		

99

# 1 Introduction

100 The *SAML V2.0 Holder-of-Key Assertion Profile* describes the issuing and processing of a holder-of-key  
 101 SAML assertion, that is, an assertion containing a `<saml:SubjectConfirmation>` element whose  
 102 Method attribute is set to `urn:oasis:names:tc:SAML:2.0:cm:holder-of-key`. Specifically, we  
 103 describe the structural characteristics of a `<ds:KeyInfo>` element with bound X.509 data and show how  
 104 a relying party confirms that such a `<ds:KeyInfo>` element matches given X.509 data. The binding  
 105 material used by the SAML issuer and the matching data used by the relying party are obtained from a  
 106 standard X.509 v3 certificate Error: Reference source not found.

107 This profile involves a SAML issuer and a SAML relying party, each with an X.509 v3 certificate in its  
 108 possession. The SAML issuer uses its certificate to produce a holder-of-key SAML assertion. The  
 109 relying party consumes the assertion, confirming the subject by comparing the X.509 data in the  
 110 assertion with the X.509 data in its possession.

## 1.1 Notation

112 This specification uses normative text.

113 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD  
 114 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as  
 115 described in [RFC2119]:

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

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

121           Listings of XML schemas appear like this.

122           Example code listings appear like this.

124 Conventional XML namespace prefixes are used throughout the listings in this specification to stand for  
 125 their respective namespaces as follows, whether or not a namespace declaration is present in the  
 126 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].

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

## 129 1.2 Normative References

- 130 [RFC2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF  
131 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>
- 132 [RFC4514] K. Zeilenga. *Lightweight Directory Access Protocol (LDAP): String*  
133 *Representation of Distinguished Names*. IETF RFC 4514, June 2006.  
134 <http://www.ietf.org/rfc/rfc4514.txt>
- 135 [RFC5280] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. *Internet*  
136 *X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL)*  
137 *Profile*. IETF RFC 5280, May 2008. <http://www.ietf.org/rfc/rfc5280.txt>
- 138 [SAML2Core] S. Cantor, J. Kemp, R. Philpott, E. Maler. *Assertions and Protocols for the*  
139 *OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard,  
140 March 2005. <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>
- 141 [SAML2Prof] J. Hughes, S. Cantor, J. Hodges, F. Hirsch, P. Mishra, R. Philpott, E. Maler.  
142 *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*.  
143 OASIS Standard, March 2005. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)  
144 [open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
- 145 [Schema1] H. S. Thompson et al. *XML Schema Part 1: Structures*. World Wide Web  
146 Consortium Recommendation, May 2001. See [http://www.w3.org/TR/2001/REC-](http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/)  
147 [xmlschema-1-20010502/](http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/)
- 148 [XMLSig] D. Eastlake, J. Reagle, D. Solo, F. Hirsch, T. Roessler. *XML Signature Syntax*  
149 *and Processing (Second Edition)*. World Wide Web Consortium  
150 Recommendation, 10 June 2008. <http://www.w3.org/TR/xmlsig-core/>

## 151 1.3 Non-normative References

- 152 [RFC3820] S. Tuecke, V. Welch, D. Engert, L. Pearlman, M. Thompson. *Internet X.509*  
153 *Public Key Infrastructure (PKI) Proxy Certificate Profile*. IETF RFC 3820, June  
154 2004. <http://www.ietf.org/rfc/rfc3820.txt>
- 155 [RFC4346] T. Dierks, E. Rescorla. *The Transport Layer Security (TLS) Protocol*. IETF  
156 RFC 4346, April 2006. <http://www.ietf.org/rfc/rfc4346.txt>

## 157 1.4 Conformance

### 158 1.4.1 SAML V2.0 Holder-of-Key Assertion Profile

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

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

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

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

## 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](mailto: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

Suppose a SAML response issued by a SAML issuer contains one or more holder-of-key assertions (otherwise this specification is not applicable). 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 v3 certificate to the <saml:SubjectConfirmation> element of a holder-of-key assertion.

The complementary process involves a relying party that confirms that the X.509 data bound to the assertion matches the data in a given X.509 v3 certificate. 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.

It is assumed that both the SAML issuer and the relying party each possess an X.509 v3 certificate that is known to be associated with the subject of the assertion. How the X.509 certificate is obtained, however, is completely out of scope.

### 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. That said, this specification mandates that the X.509 data bound to the SAML assertion by the SAML issuer and matched against the bound X.509 data by the relying party MUST be taken from an X.509 v3 certificate. Error: Reference source not found. The specific characteristics of these certificates, however, 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 therefore all portions of the certificate, apart from the X.509 data specified in the following sections, are unspecified.

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

In what follows, we use the term *X.509 certificate* to refer to an X.509 v3 certificate conforming to Error: Reference source not found.

## 208 2.4 Issuing Holder-of-Key Assertions

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

212 Suppose a SAML issuer wishes to issue a response containing one or more holder-of-key assertions. As  
213 a prerequisite, the SAML issuer MUST possess an X.509 certificate known to be associated with the  
214 subject. The SAML issuer binds some or all of the X.509 data in the certificate to the  
215 `<saml:SubjectConfirmation>` element of a SAML assertion. The expected content of a holder-of-  
216 key `<saml:SubjectConfirmation>` element is specified in the next section.

217 The SAML issuer binds a `<ds:KeyInfo>` element to a SAML assertion. The `<ds:KeyInfo>` element  
218 contains one or more of the following elements: `<ds:X509Certificate>`, `<ds:X509SKI>`,  
219 `<ds:X509SubjectName>`, or `<ds:X509IssuerSerial>`. The `<ds:X509Certificate>` element  
220 contains a base64 encoding of the certificate possessed by the SAML issuer. The `<ds:X509SKI>`  
221 element contains the base64 encoding of the Subject Key Identifier (SKI) extension (if there is one)  
222 bound to the certificate. The `<ds:X509SubjectName>` element contains the subject distinguished  
223 name (DN) bound to the certificate. The `<ds:X509IssuerSerial>` element contains the issuer DN  
224 and the issuer serial number bound to the certificate. In each case, the content of the `<ds:KeyInfo>`  
225 element conforms to the XML Signature specification [XMLSig]. These requirements are spelled out  
226 more clearly in the next section.

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

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

### 237 2.4.1 KeyInfo Usage

238 According to the SAML V2.0 specification, a holder-of-key `<saml:SubjectConfirmation>` element  
239 MUST contain at least one `<ds:KeyInfo>` element and that the `<ds:KeyInfo>` element conform to  
240 the XML Signature specification. The current specification requires that the `<ds:KeyInfo>` element  
241 MUST conform to the *Second Edition* of the XML Signature specification [XMLSig] and further constrains  
242 the content of each `<ds:KeyInfo>` element to contain exactly one `<ds:X509Data>` element. The  
243 `<ds:X509Data>` element MUST NOT contain a `<ds:X509CRL>` element. Instead, the following content  
244 options are specified, at least one of which MUST be satisfied:

- 245 • The `<ds:X509Data>` element MAY contain a `<ds:X509Certificate>` element. If it does, the  
246 `<ds:X509Certificate>` element MUST contain a base64 encoding of the X.509 certificate  
247 possessed by the SAML issuer.
- 248 • The `<ds:X509Data>` element MAY contain a `<ds:X509SKI>` element. If it does, the  
249 `<ds:X509SKI>` element MUST contain the base64 encoding of the plain (i.e., *not* DER-  
250 encoded) value of the Subject Key Identifier (SKI) extension (as specified in [XMLSig]) of the  
251 X.509 certificate possessed by the SAML issuer. If the certificate does not contain an SKI  
252 extension, the `<ds:X509Data>` element MUST NOT contain a `<ds:X509SKI>` element.

253 • The <ds:X509Data> element MAY contain a <ds:X509SubjectName> element. If it does, the  
254 <ds:X509SubjectName> element MUST contain the subject distinguished name (DN) bound  
255 to the X.509 certificate possessed by the SAML issuer.

256 • The <ds:X509Data> element MAY contain a <ds:X509IssuerSerial> element. If it does,  
257 the <ds:X509IssuerSerial> element MUST contain the issuer DN and the issuer serial  
258 number (as specified in [XMLSig]) bound to the X.509 certificate possessed by the SAML issuer.

259 Use of the <ds:X509Certificate> element or the <ds:X509IssuerSerial> element is most  
260 restrictive since each implies that the exact same certificate is used by both the SAML issuer and the  
261 relying party. Use of the <ds:X509SKI> element or the <ds:X509SubjectName> element is less  
262 restrictive since each permits a different certificate to be used by the relying party provided the certificate  
263 contains the same key or DN (resp.) used by the SAML issuer.

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

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

## 271 2.4.2 Example

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

```
274 <saml:SubjectConfirmation
275   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
276   Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
277   <saml:SubjectConfirmationData
278     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
279     xsi:type="saml:KeyInfoConfirmationDataType">
280     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
281       <ds:X509Data>
282
283         <!-- a base64 encoding of an X.509 certificate -->
284         <ds:X509Certificate>
285 MIIDuDCCAqACCQJZK8wF0xVXjANBqkqhkiG9w0BAQQFADCBnTELMaKGA1UEBhMCQlIxIExEzARBgNV
286 BAgTC1NvbWUtu3RhdGUxEjAQBGNVBACTCVNvbWUtuQ210eTESMBAGA1UEChMJR1NvQyAyMDA4MRIw
287 EAYDVQQLLEw1HU29DIDIwMDgxFzAVBGNVBAMTDkpvYW5hIFRyaW5kYWRLMSQwIgyJKoZIhvcNAQkB
288 FhVzb211LWFkZHZJLc3NAAG9zdC5vcmcwHhcNMDgwNjE2MTcyMTQzWhcNMDkwNjE2MTcyMTQzWjCB
289 nTELMaKGA1UEBhMCQlIxIExEzARBgNVBAGTC1NvbWUtu3RhdGUxEjAQBGNVBACTCVNvbWUtuQ210eTES
290 MBAGA1UEChMJR1NvQyAyMDA4MRIwEAYDVQQLLEw1HU29DIDIwMDgxFzAVBGNVBAMTDkpvYW5hIFRya
291 aW5kYWRLMSQwIgyJKoZIhvcNAQkBFhVzb211LWFkZHZJLc3NAAG9zdC5vcmcwggEiMA0GCSqGSIB3
292 DQEBAAQUAA4IBDwAwggEKAoIBAQDIDVkd0CCVYA0TspOPmcSNnivjQq7jCacrgRPawKi3/pTuvnW
293 3c2XCpyT2s6Sks3Eg5T4HIXta5E+1OpN8VbTunVdSrac54r2uK8x+8AqX7M0wQw+98iGw9E2an5q
294 xRZfqgE1T5jWL/a/G1/e2TGImp521W3k1nNtf8rYH39JpwBSZMeW7uHOSZokT/pVvqPTgG7vUQT6
295 BiRh7PfwLrLOMubbeQ6Z2m3Vnsv20E1FbPzswszh4X1gXj9bnyI2UsuoisW9Y4p4byjL3GJ/hxp
296 mjRjXs+aIpzi0V3MH+jVJ98eomhlUFLaE83xycC8lns+FcCSQZ8RsbnaLZrtC8r7AgMBAAEwDQYJ
297 KoZIhvcNAQEEBQADggEBACwnWSEpwq5aE7QBdDNNXyok34RIonYi9690yw7i+JU7R/QdE42GERJS
298 DVKBN959ELLJf5d0vybGv08QWbZVQ7eBGn9xaZ7MhSnblyNDXs9vuv1V2Dy32q1J5nCSzqpJDyln
299 lVFWe9UQMCJOO6ibUtWLhIDQ49kmMabgyYfX28qB6oRdVL+mDI/XTt+mkCgk4Rs78n4kbX6qnRlj
300 dE/YnibP1A7iMh8pQkv49J6sP9SeUmQ2zXKct3tSRzzyWc8JjOZGuBYGQH19Xm7WEs4CXS7izJW
301 E32frMatavMcTM/gnDtCc8tZAx12PSLOF1954vapfMjBhg3VTI6QRW//wPE=
302 </ds:X509Certificate>
303
304 <!-- the above X.509 certificate does not contain a
305 Subject Key Identifier extension so the SAML issuer
306 must not include a <ds:X509SKI> element -->
307
```



```

308     <!-- the subject DN (in RFC 5414 format) bound to the
309         above X.509 certificate -->
310     <ds:X509SubjectName>emailAddress=some-address@host.org,CN=Joana
311 Trindade,OU=GSoc 2008,O=GSoc 2008,L=Some-City,ST=Some-
312 State,C=BR</ds:X509SubjectName>
313
314     <!-- the issuer DN (in RFC 5414 format) and the issuer serial
315         number (in decimal) bound to the above X.509 certificate -->
316     <ds:X509IssuerSerial>
317         <ds:X509IssuerName>emailAddress=some-address@host.org,CN=Joana
318 Trindade,OU=GSoc 2008,O=GSoc 2008,L=Some-City,ST=Some-
319 State,C=BR</ds:X509IssuerName>
320         <ds:X509SerialNumber>9900230501951362398</ds:X509SerialNumber>
321     </ds:X509IssuerSerial>
322
323     </ds:X509Data>
324 </ds:KeyInfo>
325 </saml:SubjectConfirmationData>
326 </saml:SubjectConfirmation>

```

327 A relying party can confirm the subject by the matching the available X.509 data to any of the above  
328 child elements of the `<ds:X509Data>` element.

## 329 2.5 Processing Holder-of-Key Assertions

330 A relying party wishing to confirm the subject of a holder-of-key assertion MUST possess an X.509  
331 certificate known to be associated with the presenter of the assertion. The relying party confirms the  
332 subject of the assertion by comparing the X.509 data in the certificate to the X.509 data bound to the  
333 assertion. If the X.509 data in the certificate matches the X.509 data bound to the assertion, the subject  
334 is said to be *confirmed*.

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

340 If the `<ds:X509Data>` element contains multiple child elements, the relying party may confirm the  
341 subject based on any one of them. Specifically, the relying party MUST confirm that the certificate  
342 matches the content of the `<ds:X509Data>` element as follows:

- 343 • If the `<ds:X509Data>` element contains a `<ds:X509Certificate>` element, and the relying  
344 party chooses to confirm the subject based on this element, the relying party MUST ensure that  
345 the certificate bound to the assertion matches the X.509 certificate in its possession. Matching is  
346 done by comparing the base64-decoded certificates, or the hash values of the base64-decoded  
347 certificates, byte-for-byte.
- 348 • If the `<ds:X509Data>` element contains a `<ds:X509SKI>` element, and the relying party  
349 chooses to confirm the subject based on this element, the relying party MUST ensure that the  
350 value bound to the assertion matches the Subject Key Identifier (SKI) extension bound to the  
351 X.509 certificate. Matching is done by comparing the base64-decoded SKI values byte-for-byte.  
352 If the X.509 certificate does not contain an SKI extension, the subject is not confirmed and the  
353 relying party SHOULD disregard the assertion.
- 354 • If the `<ds:X509Data>` element contains a `<ds:X509SubjectName>` element, and the relying  
355 party chooses to confirm the subject based on this element, the relying party MUST ensure that  
356 the subject distinguished name (DN) bound to the assertion matches the DN bound to the X.509  
357 certificate. If, however, the relying party does not trust the certificate issuer to issue such a DN,  
358 the subject is not confirmed and the relying party SHOULD disregard the assertion.

- 359       • If the `<ds:X509Data>` element contains a `<ds:X509IssuerSerial>` element, and the relying  
360 party chooses to confirm the subject based on this element, the relying party MUST ensure that  
361 the issuer DN and issuer serial number bound to the assertion match the issuer DN and the  
362 issuer serial number (resp.) bound to the X.509 certificate. If the relying party does not trust the  
363 certificate issuer to issue X.509 certificates, however, the subject is not confirmed and the relying  
364 party SHOULD disregard the assertion.

365 In the case of a `<ds:X509Certificate>` element or a `<ds:X509SKI>` element, the matching process  
366 is relatively straightforward. If the `<ds:X509Data>` element contains a `<ds:X509SubjectName>`  
367 element or a `<ds:X509IssuerSerial>` element, however, and the relying party chooses to confirm  
368 the subject based on one of these elements, the relying party MUST trust the issuer of the X.509  
369 certificate before the subject can be considered confirmed. If such a trust relationship between the  
370 relying party and the certificate issuer does not exist, the relying party SHOULD disregard the enclosing  
371 assertion.

372 If the `<saml:SubjectConfirmationData>` element includes `NotBefore` or `NotOnOrAfter`  
373 attributes, and the relying party trusts the issuer of the X.509 certificate, the relying party MUST confirm  
374 that the current time is greater than or equal to (resp., less than or equal to) the value of the `NotBefore`  
375 (resp., the `NotOnOrAfter`) attribute. If this requirement is not met, the subject is not confirmed and the  
376 relying party SHOULD disregard the assertion.

## 377 2.6 Security and Privacy Considerations

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

387 Deployments that rely on holder-of-key SAML assertions will no doubt impose their own requirements on  
388 the X.509 certificates used to obtain those assertions. For example, some deployments will require the  
389 certificate to be an X.509 end-entity certificate Error: Reference source not found issued by a trusted  
390 X.509 certification authority (CA) or a certificate based on a trusted X.509 end-entity certificate (such as  
391 an X.509 proxy certificate [RFC3820]). This specification imposes no such restrictions, however.

### 392 2.6.1 ASN.1 Encoding

393 For compatibility with the XML Signature specification [XMLSig], this profile intentionally avoids any  
394 discussion of the ASN.1 encoding of the X.509 certificates possessed by the SAML issuer and the relying  
395 party. Indeed, in the case of the `<ds:X509Certificate>` element, the ASN.1 encoding of the  
396 certificate doesn't matter. In this case, the SAML issuer simply base64-encodes the ASN.1-encoded  
397 certificate in its possession and binds it to the `<ds:X509Certificate>` element. Later the relying  
398 party base64-decodes the content of the `<ds:X509Certificate>` element and compares the resulting  
399 certificate (byte-for-byte) with the ASN.1-encoded certificate in its possession.

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

406 The problem is that the ASN.1 encoding of an X.509 certificate is not specified. While it is true that an  
407 X.509 certificate is often DER-encoded, a robust implementation must be prepared to handle other  
408 ASN.1 encodings besides DER, mainly BER and CER. Consequently it is anticipated that deployments  
409 will prefer the `<ds:X509Certificate>` element for maximum interoperability. In fact, this preference  
410 is reflected in the conformance requirements of this profile (section 1.4).

## 411 **2.6.2 X.509 Serial Number**

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

## 417 **Appendix A. Acknowledgments**

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

- 420 • TBD

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

- 422 • Joana M. F. da Trindade, Universidade Federal do Rio Grande do Sul (Brazil)
- 423 • The members of the IETF PKIX Working Group
- 424 • [Peter Sylvester, EdelWeb](#)

## 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
<a href="#">sstc-saml2-holder-of-key-draft-07</a>	<a href="#">7 Dec 2008</a>	<a href="#">T. Scavo</a>	<a href="#">Added NotBefore/NotOnOrAfter attributes</a>