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

## Working Draft 04~~03~~, ~~7 September~~6 October 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 public key 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.

# Notices



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

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

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

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

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

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

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

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

# Table of Contents

# 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 a standard X.509 public key certificate.

## 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]. |

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

## 1.2 Normative References


**[RFC2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt

**[RFC4514]** K. Zeilenga. *Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names.* IETF RFC 4514, June 2006. http://www.ietf.org/rfc/rfc4514.txt

**[RFC5280]** D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF RFC 5280, May 2008. http://www.ietf.org/rfc/rfc5280.txt

**[SAML2Core]** S. Cantor, J. Kemp, R. Philpott, E. Maler. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0.* OASIS Standard, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

**[SAML2Prof]** J. Hughes, S. Cantor, J. Hodges, F. Hirsch, P. Mishra, R. Philpott, E. Maler. *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0.* OASIS Standard, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

**[XMLSig]** D. Eastlake, J. Reagle, D. Solo, F. Hirsch, T. Roessler. *XML Signature Syntax and Processing (Second Edition).* World Wide Web Consortium Recommendation, 10 June 2008. http://www.w3.org/TR/xmldsig-core/

## 1.3 Non-normative References

**[AIXCM]** T. Moreau. *Auto Issued X.509 Certificate Mechanism (AIXCM).* IETF Internet-Draft, 6 August 2008. See http://www.ietf.org/internet-drafts/draft-moreau-pkix-aixcm-00.txt

**[RFC3820]** S. Tuecke, V. Welch, D. Engert, L. Pearlman, M. Thompson. *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile.* IETF RFC 3820, June 2004. http://www.ietf.org/rfc/rfc3820.txt

**[RFC4346]** T. Dierks, E. Rescorla. *The Transport Layer Security (TLS) Protocol.* IETF RFC 4346, April 2006. http://www.ietf.org/rfc/rfc4346.txt

~~**[RFC5280]** D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.* IETF RFC 5280, May 2008. http://www.ietf.org/rfc/rfc5280.txt~~

## 1.4 Conformance

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

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

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

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

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

**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:** ~~Extends~~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 public key 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 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 have access to an X.509 public key 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) available to 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 MUST be taken from an X.509 public key certificate. Likewise the X.509 data matched against the bound X.509 data by the relying party MUST also be taken from an X.509 public key certificate. 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 out of scope.

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.

## 2.4 Issuing Holder-of-Key Assertions

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

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

### 2.4.1 KeyInfo Usage

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

- The <ds:X509Data> element MAY contain a <ds:X509Certificate> element. If it does, the <ds:X509Certificate> element MUST contain a base64 encoding of a DER-encoded X.509 certificate.
- The <ds:X509Data> element MAY contain a <ds:X509SKI> element. If it does, the <ds:X509SKI> element MUST contain ~~a~~the base64 encoding of the SHA-1 hash of the public key bound to an X.509 certificate.
- The <ds:X509Data> element MAY contain a <ds:X509SubjectName> element. If it does, the <ds:X509SubjectName> element MUST contain the subject distinguished name (DN) bound to an X.509 certificate.
- The <ds:X509Data> element MAY contain a <ds:X509IssuerSerial> element. If it does, the <ds:X509IssuerSerial> element MUST contain the issuer DN and the issuer serial number (as specified in [XMLSig]) bound to an X.509 certificate.

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

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

The unencoded value of the <ds:X509SKI> element is the same value a certificate issuer would compute for the Subject Key Identifier extension of the certificate (see section 4.2.1.2 of [RFC5280]). Specifically, the value of the <ds:X509SKI> element is computed as follows:

1. Extract the SubjectPublicKeyInfo field from the certificate.
2. Parse the SubjectPublicKeyInfo field and extract the BIT STRING subjectPublicKey.
3. Compute the 160-bit SHA-1 hash over the BIT STRING subjectPublicKey.

4. Compute the base64 encoding of the SHA-1 hash.

The preceding algorithm produces a 28-character string value for the <ds:X509SKI> element.

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

## 2.4.2 Example

Here is an example of a holder-of-key <saml:SubjectConfirmation> element illustrating each of the ~~containing both a <ds:X509Certificate> element and a <ds:X509SKI> element~~content options specified in section 2.4:

```
<saml:SubjectConfirmation
  Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
  <saml:SubjectConfirmationData
    xsi:type="saml:KeyInfoConfirmationDataType">
    <ds:KeyInfo>
      <ds:X509Data>

        <!-- a base64 encoding of a DER-encoded X.509 certificate -->
        <ds:X509Certificate>
MIIDuDCCAqACCQCJZK8wF0xVXjANBgkqhkiG9w0BAQQFADCBnTELMAkGA1UEBhMCQlIxEzARBgNV
BAgTClNvbWUtU3RhdGUxEjAQBgNVBAcTCVNvbWUtQ2l0eTESMBAGA1UEChMJR1NvQyAyMDA4MRIw
EAYDVQQLEwlHU29DIDIwMDgxFzAVBgNVBAMTDkpvYW5hIFRyaW5kYWRlMSQwIgYJKoZIhvcNAQkB
FhVzb21lLWFkZHJlc3NAaG9zdC5vcmcwHhcNMDgwNjE2MTcyMTQzWhcNMDkwNjE2MTcyMTQzWjCB
nTELMAkGA1UEBhMCQlIxEzARBgNVBAgTClNvbWUtU3RhdGUxEjAQBgNVBAcTCVNvbWUtQ2l0eTES
MBAGA1UEChMJR1NvQyAyMDA4MRIwEAYDVQQLEwlHU29DIDIwMDgxFzAVBgNVBAMTDkpvYW5hIFRy
aW5kYWRlMSQwIgYJKoZIhvcNAQkBFhVzb21lLWFkZHJlc3NAaG9zdC5vcmcwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDIDVKdO2CCVYA0TspOPmcSNnivjQq7jCacrgRPawKi3/pTuvnW
3c2XCpyT2s6Sks3Eg5T4HIXta5E+lOpN8VbTunVdSrac54r2uK8x+8AqX7M0wQw+98iGw9E2an5q
xRZfqqE1T5jWL/a/G1/e2TGlmp521W3k1nNtf8rYH39JpwBSZMeW7uHOSZOkT/pVvqPTgG7vUQT6
BiRh7PfwsLrLOMubbeQ6Z2m3Vnsv20E1FbPzwswzh4X1gXj9bnyI2UsuoisW9Y4p4byjL3GJ/hxp
mjRjXs+aIpzi0V3MH+jVJ98eomhlUFLaE83xycC8lns+FcCSQZ8RsbnaLZrtC8r7AgMBAAEwDQYJ
KoZIhvcNAQEEBQADggEBACwnWSEpwq5aE7QBdDNNXyok34RIonYi9690yw7i+JU7R/QdE42GERJS
DVKBN959ELLJf5d0vybGv08QWbZVQ7eBGn9xaZ7MhSnblYNDXs9vuv1V2Dy32q1J5nCSzqpJDyln
lVFWe9UQMCJOO6ibUtWLhiDQ49kmMabgyYfX28qB6oRdVL+mDI/XTt+mkCgk4Rs78n4kbX6qnRlj
dE/YnibP1A7iMh8pQkv49J6sP9SeUmQ2zxKCt3tSRzzypWc8JjOZGuBYGQHl9Xm7WEs4CXS7iZJW
E32frMAtavMcTM/gnDtCc8tZAxl2PSLOF1954vapfMjBhg3VTI6QRW//wPE=
        </ds:X509Certificate>

        <!-- the base64 encoding of the SHA-1 hash of the
             public key bound to the above X.509 certificate -->
        <ds:X509SKI>YphoxnLNAx/S0sdbdN3nD01wuR8=</ds:X509SKI>

        <!-- the subject DN (in RFC 5414 format) bound to the
             above X.509 certificate -->
        <ds:X509SubjectName>emailAddress=some-address@host.org,CN=Joana
Trindade,OU=GSoC 2008,O=GSoC 2008,L=Some-City,ST=Some-
State,C=BR</ds:X509SubjectName>

        <!-- the issuer DN (in RFC 5414 format) and the issuer serial
             number (in decimal) bound to the above X.509 certificate -->
        <ds:X509IssuerSerial>
          <ds:X509IssuerName>emailAddress=some-address@host.org,CN=Joana
Trindade,OU=GSoC 2008,O=GSoC 2008,L=Some-City,ST=Some-
State,C=BR</ds:X509IssuerName>
          <ds:X509SerialNumber>9900230501951362398</ds:X509SerialNumber>
        </ds:X509IssuerSerial>

      </ds:X509Data>
    </ds:KeyInfo>
  </saml:SubjectConfirmationData>
```

```
</saml:SubjectConfirmation>
```

~~Note that the key in the <ds:X509SKI> element is in fact an alternate representation of the public key bound to the certificate in the <ds:X509Certificate> element.~~ A relying party can confirm the subject by the matching the available X.509 data to ~~either~~any of ~~these~~the above child elements.

## 2.5 Processing Holder-of-Key Assertions

A relying party wishing to confirm the subject of a holder-of-key assertion MUST have access to an X.509 public key certificate known to be associated with the ~~subject~~presenter of the assertion. The relying party confirms the subject of the assertion by comparing the X.509 data in the certificate to the X.509 data bound to the assertion. If the X.509 data in the certificate matches the X.509 data bound to the assertion, the subject is said to be *confirmed*.

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

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

- If the <ds:X509Data> element contains a <ds:X509Certificate> element, and the relying party chooses to confirm the subject based on this element, the relying party MUST confirm that the DER-encoded certificate bound to the assertion matches the X.509 certificate. Matching is done by comparing the certificates, or the hash values of the certificates, byte-for-byte.
- If the <ds:X509Data> element contains a <ds:X509SKI> element, and the relying party chooses to confirm the subject based on this element, the relying party MUST confirm that the hash value bound to the assertion matches the SHA-1 hash of the public key bound to the X.509 certificate~~-~~.
- If the <ds:X509Data> element contains a <ds:X509SubjectName> element, and the relying party chooses to confirm the subject based on this element, the relying party MUST confirm that the subject distinguished name (DN) bound to the assertion matches the DN bound to the X.509 certificate. If, however, the relying party does not trust the certificate issuer to issue such a DN, the subject is not confirmed and the relying party SHOULD disregard the enclosing assertion.
- If the <ds:X509Data> element contains a <ds:X509IssuerSerial> element, and the relying party chooses to confirm the subject based on this element, the relying party MUST confirm that the issuer DN and issuer serial number bound to the assertion match the issuer DN and the issuer serial number (resp.) bound to the X.509 certificate. If the relying party does not trust the certificate issuer to issue X.509 certificates, however, the subject is not confirmed and the relying party SHOULD disregard the enclosing assertion.

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

## 2.6 Security and Privacy Considerations

This profile assumes that both the SAML issuer and the relying party have access to an X.509 public key certificate. For those deployments that wish to avoid or do not require a public key infrastructure (PKI), this

may seem unnecessarily restrictive. In fact, the use of X.509 certificates is typical and provides a number of advantages. First, ~~if the subject is the SAML requester, the subject DN of the certificate may be used in lieu of an URI. Second,~~ observe that the SSL/TLS protocol [RFC4346] requires the use of X.509 certificates. Second~~Finally~~, and most importantly, since there is no presumption of an underlying trust model for X.509 certificates, the full range of possible content for the `<ds:KeyInfo>` element is avoided. Those deployments that are in fact based on such a trust model, or wish to avoid X.509 certificates altogether, may choose to profile additional child elements such as `<ds:KeyName>` or `<ds:KeyValue>`.

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

In particular, note that self-signed certificates are permitted with this specification. However, self-signed certificates should be used with care since it is well known that the use of such certificates may break certain implementations or protocols. For maximum interoperability, implementers are encouraged to use X.509 end-entity certificates [RFC5280] whenever possible. For those deployments that wish to avoid or do not require a PKI, yet want to maintain interoperability, observe that so-called "meaningless X.509 certificates" [AIXCM] satisfy the requirements of X.509 end-entity certificates without belaboring the assumption of an underlying trust model.

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

# Appendix A. Acknowledgments

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

- TBD

The editor would also like to acknowledge the following contributors:

- Joana M. F. da Trindade, Universidade Federal do Rio Grande do Sul (Brazil)

# 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 `samlp:` |
| 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 |