Workgroup: Network Working Group
Internet-Draft: draft-santesson-svt-xml-03
Published: 21 March 2022
Intended Status: Informational
Expires: 22 September 2022
Authors: S. Santesson R. Housley
IDsec Solutions Vigil Security

XML Signature Validation Token

Abstract

This document defines a XML profile for the Signature Validation Token defined in [SVT].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 September 2022.

Table of Contents

1. Introduction

The "Signature Validation Token" specification [SVT] defines the basic token to support signature validation in a way that can significantly extend the lifetime of a signature.

This specification defines a profile for implementing SVT with a signed XML document, and defines the following aspects of SVT usage:

- How to include reference data related to XML signatures and XML documents in an SVT.

- How to add an SVT token to a XML signature.

XML documents can have any number of signature elements, signing an arbitrary number of fragments of XML documents. The actual signature element may be included in the signed XML document (enveloped), include the signed data (enveloping) or may be separate from the signed content (detached).

To provide a generic solution for any type of XML signature an SVT is added to each XML signature element within the XML signature <ds:Object> element.

2. Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The definitions in [SVT] apply also to this document.

2.1. Notation

2.1.1. References to XML Elements from XML Schemas

When referring to elements from the W3C XML Signature namespace (http://www.w3.org/2000/09/xmldsig#) the following syntax is used:

*<ds:Signature>

When referring to elements from the ETSI XAdES XML Signature namespace (http://uri.etsi.org/01903/v1.3.2#) the following syntax is used:

*<xades:CertDigest>

When referring to elements defined in this specification (http://id.swedenconnect.se/svt/1.0/sig-prop/ns) the following syntax is used:

*<svt:Element>

3. SVT in XML Documents

When SVT is provided for XML signatures then one SVT MUST be provided for each XML signature.

An SVT embedded within the XML signature element MUST be placed in a <svt:SignatureValidationToken> element as defined in Section 3.1.

3.1. SignatureValidationToken Signature Property

The <svt:SignatureValidationToken> element MUST be placed in a <ds:SignatureProperty> element in accordance with [XMLDSIG11]. The <ds:SignatureProperty> element MUST be placed inside a <ds:SignatureProperties> element inside a <ds:Object> element inside a <ds:Signature> element.

Note: [XMLDSIG11] requires the Target attribute to be present in <ds:SignatureProperty>, referencing the signature targeted by this signature property. If an SVT is added to a signature that do not have an Id attribute, implementations SHOULD add an Id attribute to the <ds:Signature> element and reference that Id in the Target attribute. This Id attribute and Target attribute value matching is required by the [XMLDSIG11] standard, but it is redundant in the context of SVT validation as the SVT already contains information that uniquely identifies the target signature. Validation applications SHOULD not reject an SVT token because of Id and Target attribute mismatch, and MUST rely on matching against signature using signed information in the SVT itself.

The <svt:SignatureValidationToken> element is defined by the following XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    targetNamespace="http://id.swedenconnect.se/svt/1.0/sig-prop/ns"
    xmlns:svt="http://id.swedenconnect.se/svt/1.0/sig-prop/ns">

  <xs:element name="SignatureValidationToken"
      type="svt:SignatureValidationTokenType" />

  <xs:complexType name="SignatureValidationTokenType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

</xs:schema>
```

The SVT token MUST be included as a string representation of the SVT JWT. Note that this is the string representation of the JWT without further encoding. The SVT MUST NOT be represented by the Base64 encoded bytes of the JWT string.

Example:

```
<ds:Signature Id="MySignatureId">
  ...
  <ds:Object>
    <ds:SignatureProperties>
      <ds:SignatureProperty Target="#MySignatureId">
        <svt:SignatureValidationToken>
              eyJ0eXAiOiJKV1QiLCJhb...2aNZ
        </svt:SignatureValidationToken>
      </ds:SignatureProperty>
    </ds:SignatureProperties>
  </ds:Object>
</ds:Signature>
```

3.2. Multiple SVT in a signature

If a new SVT is stored in a signature which already contains a previously issued SVT, implementations can choose to either replace the existing SVT or to store the new SVT in addition to the existing SVT.

If the new SVT is stored in addition to the old SVT, it SHOULD be stored in a new <ds:SignatureProperty> element inside the existing <ds:SignatureProperties> element where the old SVT is located.

For interoperability robustness, signature validation applications MUST be able to handle signatures where the new SVT is located in a new <ds:Object> element.

4. SVT Claims

4.1. Profile Identifer

When this profile is used the SigValidation object MUST contain a "profile" claim with the value "XML".

4.2. Signature Reference Data

The SVT Signature object MUST contain a "sig_ref" claim (SigReference object) with the following elements:

* "id" -- The Id-attribute of the XML signature, if present.

* "sig_hash" -- The hash over the signature value bytes.

* "sb_hash" -- The hash over the canonicalized <ds:SignedInfo> element (the bytes the XML signature algorithm has signed to generated the signature value).

4.3. Signed Data Reference Data

The SVT Signature object MUST contain one instance of the "sig_data" claim (SignedData object) for each <ds:Reference> element in the <ds:SignedInfo> element. The "sig_data" claim MUST contain the following elements:

* "ref" -- The value of the URI attribute of the corresponding <ds:Reference> element.

* "hash" -- The hash of all bytes identified corresponding <ds:Reference> element after applying all identified canonicalization and transformation algorithms. These are the same bytes that is hashed by the hash value in the <ds:DigestValue> element inside the <ds:Reference> element.

4.4. Signer Certificate References

The SVT Signature object MUST contain a "signer_cert_ref" claim (CertReference object). The "type" parameter of the "signer_cert_ref" claim MUST be either "chain" or "chain_hash".

* The "chain" type MUST be used when signature validation was performed using one or more certificates where some or all of the certificates in the chain are not present in the target signature.

* The "chain_hash" type MUST be used when signature validation was performed using one or more certificates where all of the certificates are present in the target signature.

5. JOSE Header

5.1. SVT Signing Key Reference

The SVT JOSE header must contain one of the following header parameters in accordance with [RFC7515], for storing a reference to the public key used to verify the signature on the SVT:

* "x5c" -- Holds an X.509 certificate [RFC5280] or a chain of certificates. The certificate holding the public key that verifies the signature on the SVT MUST be the first certificate in the chain.

* "kid" -- A key identifier holding the Base64 encoded hash value of the certificate that can verify the signature on the SVT. The hash algorithm MUST be the same hash algorithm used when signing the SVT as specified by the alg header parameter.

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

The security considerations of [SVT] applies also to this document.

8. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key

Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[SVT] Santesson, S. and R. Housley, "Signature Validation Token", IETF draft-santesson-svt-02, September 2021.

[XMLDSIG11] Eastlake, D., Reagle, J., Solo, D., Hirsch, F., Nystrom, M., Roessler, T., and K. Yiu, "XML Signature Syntax and Processing Version 1.1", W3C Proposed Recommendation, 11 April 2013.

Authors' Addresses

Stefan Santesson
IDsec Solutions AB
Forskningsbyn Ideon
SE-223 70 Lund
Sweden

Email: sts@aaa-sec.com

Russ Housley
Vigil Security, LLC
516 Dranesville Road
Herndon, VA, 20170
United States of America

Email: housley@vigilsec.com