XML Signature Profile

Version 1.0

Sep 2015

Controller of Certifying Authorities

Department of Information Technology

Ministry of Communications and Information Technology

Document Control

Document Name	XML Signature Profile
Status	Release
Version	1.0
Last update	10 Sep 2015
Document Owner	Controller of Certifying Authorities, India

Table of Contents

1. Definitions

"**Canonicalisation**", in relation to a xml digital signature, means the process of converting electronic record that has more than one possible representation into a 'standard', 'normal', or 'canonical form' in which the variations in representation of electronic record shall be standardised by applying consistent rules, primarily as part of the xml digital signature creation and verification processes. It is necessary that canonicalization is performed as part of the XML Digital Signature creation and verification processes in order to ensure that the signer and verifier are computing the hashes on the same string of bits.

"**counter signature**" means a signature on a previous signature in a series of signatures, affixed after the verification the signature on electronic record and subsequent signatures on previous signatures serially.

"**detached signature**" means the signature that is stored independent of electronic record being signed. Detached Signature over content is separate from the Signature element. The content may or may not reside within the same XML document containing the XML signature.

"**digestmethod element**", in relation to a xml digital signature, means the digest algorithm to be used for the original data object or transformed, if any 'xml transforms' exists;

"**digestvalue element**" means the value of the digest;

"end entity**"** means the subscriber or system on behalf of the subscriber in whose name the Electronic Signature Certificate is issued;

"**end entity signature**" means authentication of any electronic record by an end entity by means of a digital signature, electronic method or procedure in accordance with the provisions of sections 3 or 3A of the Act;

"**enveloped signature**" means enveloping of the signature and the initial electronic record into another electronic record;

"**enveloping signature**" means a signature over a electronic record that is referenced and contained within the signature element;

"**initial electronic record**", in the context of xml digital signature process, means canonicalised and transformed form of signedInfo;

"**keyinfo element**" means an element that enables key information to be packaged along with the signature element. It can contain keys, keys names, or certificates. This report recommends that KeyInfo must be present in the XML Digital Signature and that it MUST have X509 Certificate element.

"**manifest element**" ,in relation to a xml digital signature, means a structure to carry a list of reference elements processing model defined by the application. Manifest appears as a reference in Signed Info. Manifest contains one or more references. The main different between manifest and normal references is that reference validation of manifest is under application control and not part of mandatory signature verification.

"**object element**" means an optional element of xml digital signature, which is used for enveloping signature where the data object being signed is included in the xml;

"**parallel signatures**" means one or more independent signature over the same electronic record in which the ordering of the signatures is not important;

"**reference element**" ,in relation to a xml digital signature, means an element that carries a references to data objects, an optional list of transforms to be applied prior to digest (xml transforms), digestmethod and digestvalue value of referenced data objects. It has the following child elements: Transforms, Digest Method, Digest Value

"**signedinfo**" ,in relation to a xml digital signature, means an element that contains a set of information to be signed for creating an xml signature, where it shall contains references to the data object that includes the canonicalisation and signature algorithms. Signed info element has the following child elements: Canonicalization Method, Signature Method, and Reference.

"**signature**" means digital signature or xml digital signature;

"**signaturevalue**" means an element that the actual value of the digital signature;

"**signaturemethod** " means an element that contains the algorithm used for signature generation and this algorithm identifies all cryptographic functions involved in the signature generation;

"**signatureproperties**" means an element that provides a way to carry additional information about the signature, such as a time stamp or any other information which are defined by application;

"**xml**" means Extensible Markup Language that provides a standard methodology with formal syntax to identify elements of information, describe the structure of data and also to store data in an independent manner , shall have the following properties,—

(i) with xml, content and presentation are separate;
(ii) the structure of xml data in a particular context is described using either xml schema or a document type definition;
(iii) xml schema or a document type definition are stored separately from the xml document itself and can be used to validate a given xml document for conformance;

"**xml digital signature element**" means an element that defined by standard xml schema for capturing the result of a digital signature operation applied to arbitrary data in xml format , shall satisfy the following,—

(i) xml digital signature element shall exist as a standalone document or envelop the data object that it signs;
(ii) xml digital signature element shall have signedinfo, signaturevalue, keyinfo, object and has id attribute of type child elements in order in which they appear;

"xml digital signature" means the digital signature on xml electronic record;

"**xml document**" means a document with xml logical and physical structure that is used to carry data elements, composed of declarations, elements, comments, character references, and processing instructions and a physical structure composed of entities, starting with the root, or document entity;

"**xml schema**" means a set of pre-defined or user defined keywords and their attributes arranged in a structured manner, shall satisfy the following,—

(i) should be used for a particular purpose where as a schema describes the structure of an xml document and provides specification of element names that indicates which elements are allowed in an xml document, and in what combinations; and
(ii) should provide extended functionality such as data types, inheritance, and presentation rules and default values for attributes;

"xml transform" means an element that specify an optional ordered list of processing steps applied to the data objects before it was digested where the transforms include canonicalization, encoding or decoding, extensible style sheet language transformations , xpath filtering, and xml schema validation;

"**xml namespace**" means a uniform resource identifier (uri) reference where the mechanisms described in the specification are used in xml documents as element types and attribute names and also to use various xml vocabularies without having name collision. XML Namespace is identified by a URI reference using the mechanisms described in the specification RFC3986.

2. XML Signature Profile

HTML was developed as a markup language to ensure that web browsers can use it to translate and compose text, images and other data into visual or audible web pages for web users. However, with the growth of internet and World Wide Web (WWW), basic and elementary functions provided by HTML were not sufficient, and necessity of a markup language was realized, that not only meets specific functionality requirements of web-based communication, but also provides interoperability for different web server and web browser interaction.

This led to introduction of Extensible Markup Language (XML), a universal standard that provides a structure for other independent markup languages to be built from and still allow for interoperability between different web based applications and platforms. However, XML should not be technically considered as successor to HTML, as XML is mainly intended for structuring data between different platforms and applications over the World Wide Web. XML still executes the job of carrying data whereas HTML ensures designing of data, and how a given data needs to be represented. The concept of structuring of data evolved as structuring would allow data to be easily read, exchanged, and acted upon by receiving entities across any browser or web servers.

Alike any other data, XML also needs to be authenticated or signed to ensure trustworthiness. This has led to the conceptualization of XML signatures, which holds few advantages over traditional DSCs, viz-a-viz its ability to sign certain or specific data or portions of a document rather than the whole document.

With growing use of XML data, the recommended XML Signature Profile will be effective in allowing various web-based platforms to ensure that authenticated data is exchanged in

XML format. XML signature mainly consists of reference hashes, details of signer (public key, signing certificate etc.) and digital signature.

Similar to other digital signatures, XML signatures also rely on the basic step mentioned below:

(i) Data is transformed into a message digest via a reliable algorithm

(ii) Message digest is then signed and sent to specified recipient

(iii) Receiver uses the data, signature, and the public key to verify digital signatures

```
<SignatureID?>
        <SignedInfo>
                <CanonicalizationMethod/>
                <SignatureMethod/>
                <Reference URI?>
                        <Transforms>?
                        <DigestMethod>
                        <DigestValue>
                </Reference>+
        </SignedInfo>
<SignatureValue>
(<KeyInfo>
        (RetrievalMethod)
                (<X509Data>


                        (X509Certificate)

                </x509Data>)
</Keyinfo>)
(<Object ID?>)*
</Signature>
```

Figure 2 – Basic XML Signature Profile

Below table provides description of the elements / attributes used in the XML Signature syntax:

#	XML Signature Element	M/O	Description
1.	Signature ID	M	• Main element of XML Digital Signature. It can exist as a standalone XML document, or can envelop the data object that it signs
2.	SignedInfo	M	• Contains set of information to be signed for creating an XML signature; • Contains references to the data object and includes the canonicalization and signature algorithms
3.	CanonicalizationMethod	M	• Procedure for converting data, that has more than one possible representation, into a standard, normal, or canonical form
4.	SignatureMethod	M	• Specifies one attribute algorithm used for signature generation which will identify all cryptographic functions involved in the signature generation.
5.	Reference URI	O	• Uniform Resource Identifier (URI) pointer to resource. There is one or more of URI in a SignedInfo
6.	Transforms	O	• Optional list of processing steps applied to the resource's content before it was digested; • Transforms can include operations such as canonicalization, encoding/decoding (including compression/inflation), XSLT, XPath, XML schema validation or XInclude.
7.	DigestMethod	M	• Specifies digest algorithm to be used for hashing the resource referenced by the URI.
8.	DigestValue	M	• Contains value of the message digest of the resource
9.	SignatureValue	M	• Contains the actual value of the digital signature; •
10.	KeyInfo	M	• Enables key information to be packaged along with signature; • Mandatory field and must be present in the XML Digital Signature and shall have X509 Certificate element
11.	X509Data	M	• within KeyInfo contains one or more identifiers of keys or X509 certificates*
12.	X509Certificate	M	• Contains a base64-encoded[X509v3] certificate*
13.	Object ID	O	• Optional element that may occur one or more times may contain any data; • May include optional MIME type, ID, and encoding attributes*

Figure 3 – Signature Description

*Abstracted from RFC 3275

Below table provides the recommended value and specification for each of the XML attribute:

#	XML Attribute	Value and Standard Specification
1.	XML Digital Signature Standard	RFC 3275 with the following constraint i. Manifest is not permitted inside Object, ii. KeyInfo containing X509Certificate element is mandatory. iii. The Reference Processing shall use the Exclusive Canonicalization (without comments) in addition to other transforms. iv. For XML resource, XSLT shall be the last transform done to enable the rendering of the document on screen. v. For rendering of document on the screen vi. Each referenced XML resource shall be implemented using XSLT. vii. Each non XML resource shall be implemented using Mime Type attribute mentioned in the object.
2.	XML Namespace	RFC 3986
3.	Signature encoding	UTF-8 RFC 3629
4.	Signature Value Encoding	Base64 RFC 4648
5.	Reference element Digest	SHA256 FIPS 180-4
6.	Signature Algorithm	SHA256withRSA PKCS-1 Version 1.5
7.	Signature block Canonicalization	Exclusive (without comments), XML-EXC-C14N, RFC 3741. For Canonical XML: i. Canonical XML 1.0 (omits comments)http://www.w3.org/TR/2001/REC-xml-c14n-20010315 ii. Canonical XML 1.1 (omits comments) http://www.w3.org/2006/12/xml-c14n11
8.	Transform Algorithms	Exclusive (without comments), XML-EXC-C14N, RFC 3741 For Canonical XML: 1. Canonical XML 1.0 (omits comments) http://www.w3.org/TR/2001/REC-xml-c14n-20010315 2. Canonical XML 1.1 (omits comments) http://www.w3.org/2006/12/xml-c14n11 XSLT-XSL Transforms (XSLT) Version 1.0. W3C http://www.w3.org/TR/1999/REC-xslt-19991116 XPath – RFC 3653
9.	Signature Type	Enveloped or enveloping or detached (internally or externally as defined in section 2.3.2)
10.	Digital Signature Certificate	(DER) X.509 V3 issued as per interoperability guidelines
11.	Public Key Algorithms	RSA PKCS-1 Version 1.5

Figure 4 – Signature Attributes

The functioning of XML signature is based on inclusion of syntax references for digital signing. This functionality in XML allows for attached, detached, parallel and counter profiles. The difference between the attached detached, parallel and counter signature profiles is based on the relation of parent, child and sibling to the signed XML document / data.

2.1 Attached Signature

There are two methods of attaching a signature with the XML signed data. The attached signature profile for XML includes below two formats:

(i) Attached Enveloped Signature format and,
(ii) Attached Enveloping Signature format

Attached Enveloped Signature:

In Enveloping signature format, signed data is embedded within the XML signature (i.e., signed data is considered a child element). Therefore, in the attached enveloping signature profile, <Signature> element is created as a root element and the XML data is inserted into it.

In the below example, <XML document> appears under the <Signature> root element:

```
<Signature>
        <XML data object>
                        A…B…C…D…
                        …D…C…B…A
        </XML data object>
</Signature>
```

Figure 5 – Attached Enveloped Signature

Attached Enveloping Signature:

In Enveloped signature Format, XML signature is embedded within the XML data object which is to be signed (i.e. signature is considered as child element of the document). Therefore, in the attached enveloped signature profile, <Signature> element is to be created

as a child element into the XML data object. In the below example, <Signature> appears under the <XML data object>:

```
<XML data object>
        <Signature>
        </Signature>
</XML data object>
```

Figure 6 – Attached Enveloping Signature

2.2 Detached Signature Profile

In Detached XML signature Profile, XML signature is not attached to the signed XML data i.e. it is neither enveloping nor enveloped attached to the data object. This signature profile can be used when it is undesirable or impractical to physically merge the XML signature with the XML data which is to be signed.

In Detached signature profile, XML signature and data are maintained in separate files or in the same XML file, but as sibling elements. The XML signature is maintained over a content which is external to the signature element, or to be identified or referred via a URI or transform. The reference URI should be accessible by both the signer and verifier.

The Detached signature profile for XML will include below two formats:

(i) Internally Detached XML Signature Format and,

(ii) Externally Detached XML Signature Format

Internally Detached XML Signature:

In Internally Detached XML Signature Format, the signature and the data object being signed can co-exist in the same XML file, and signature can include URI references to the signed data object within the same file as sibling items.

```
<Signature>
        <SignedInfo>
                ……..
        <Reference URI=”#XYZ”>
                ……..
        </SignedInfo>
</Signature>

        <Data object>
                <XYZ Id=”xyz1”>
                    ……..
                    ……..
                </XYZ>
        </Data object>
```

Figure 7 – Internally Detached Signature

As illustrated in figure 7, <Signature> is present in the same XML file and refers to the <Data object> through a reference URI. Both, <Signature> and <Data Object> are present in the same XML file.

Externally Detached XML Signature:

In Externally Detached XML Signature Format, the signature and the signed data object will not co-exist in the same XML file or document, and can include reference URI towards an external file.

In the below example, reference URI is a web-based URL - 'http://samplereferencewebsite.com/data.xml' which is an external entity and detached from the signed document object.

```
<Signature>
        <SignedInfo>
              …….
        <Reference URI=http://samplereferencewebsite.com/data.xml>
              …….
        </SignedInfo
</Signature>
```

Figure 8 – Externally Detached Signature

This fundamental principle of internally detached profile may raise disarray with the other two formats of attached Signature profile; as in enveloping and enveloped profiles, XML signature and the XML data being signed also remain in the same XML file or document. However, the significant difference with the internally detached signature profile will be that there is no parent-child relationship between the XML signature and the XML signed data.

2.3 Counter Signature Profile

Counter Signature Profile includes signing of a previously signed data or document for authenticating the identity of the previous signature/s. In various government related and commercial activities, where counter authentication is required over a data, countersigning is considered as one of the legal mandates. Counter signature is the authoritative signature marked on a document and is exercised to establish or verify the identity of the previous signer/s. There may be single or multiple previous signatures that may be countersigned. In other words, counter signature can be recursive, i.e., can have multiple layers, with each layer attesting to the validity of the previous layers.

Counter Signature in a XML document should be asserted through attached enveloping signature. As counter signature is applied to attest a previous signature on the XML data, therefore, the signed XML data should be wrapped with the previous XML signature/s using the attached enveloping signature profile technique, and then counter signature should be applied to the object (signed XML data and the previous signature/s).

In section 2.1, Attached Enveloping Signature has been explained. In attached enveloping signature profile, signature element is the root element and the XML data to be signed is a child element. Therefore, the concerned XML data is encapsulated within the signature.

In Counter Signature Prolife, the previous XML signature/s and the XML data, on which the signature has been applied, will exist as an object that will be counter signed.

The previous signer/s can sign the data in any of the attached (enveloped or enveloping) or detached (internally or externally) format. But, to ensure that the previous signature/s , acts as the child element to the counter signature, the previous signature/s should be encapsulated with the signed data so that it formulates as an object to which counter signature will be applied along with the data.

This is illustrated in the below diagram where a previously signed XML data and the signature are together considered as child element for counter signing:

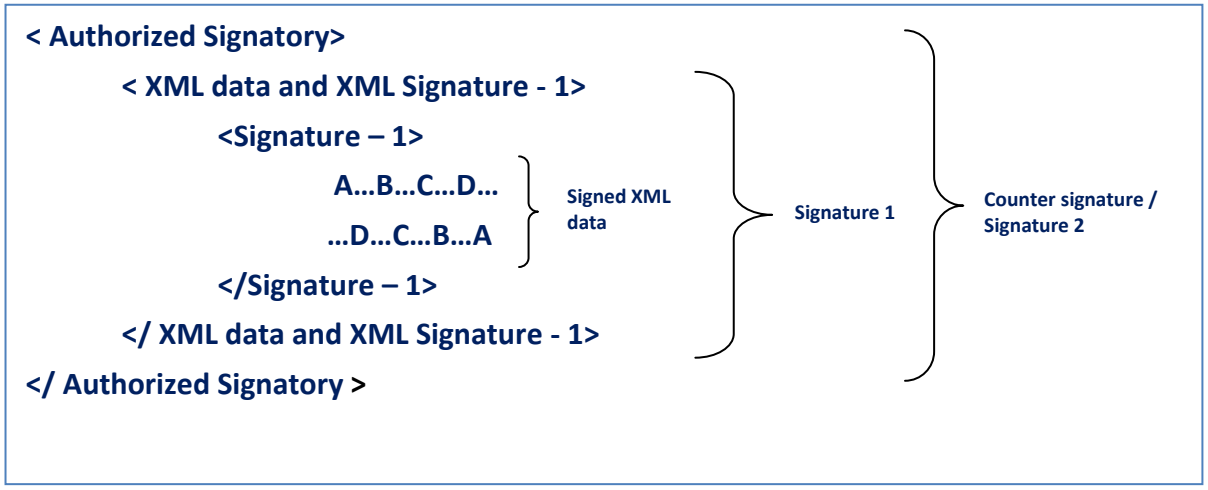

Figure 9 – Counter Signature

2.4 Parallel Signature Profile

Parallel signing includes two or more signatures on XML data. The difference between countersigning and parallel signing is that counter signature is done to verify or confirm the the previous signature/s. Counter signature is used to provide additional authority for a signature, or signal approval for another user's approval. However, parallel signing are multiple signatures required over a given data, and none of the signature is superior to the other, but their level of authority is treated as equal and they all sign the same data.

Parallel signing in XML should be done though the concept of detached signature. Ideally, it should be preferred that signatures should be internally detached, where the signatures and the object data which needs to be signed exists in the same XML file.

The internally detached is prescribed for parallel signing so that none of the signatures encapsulates or is encapsulated by the object (to be signed), but each signature includes the object as a signed reference. Figure – 10 illustrates the prescribed parallel signing profile, where two entities 'Signer1'and 'Signer2' sign the object 'agreement'.

There may be scenarios, which may imply application of both parallel and counter signing on a certain XML data. For example, counter sign may be required over a set of 3 parallel signatures. With the Parallel Signature Profile and Counter Signature Profile discussed in the respective sections, internally detached signature can be used for parallel signing and all the three parallel signatures can be counter signed by enveloping all the three parallel signatures with the XML data using attached enveloping signature technique.

```
<Data object>
  <Agreement ID>"agreement1"</Agreement ID>
  <name>Agreement1</name>
  <price>10000000</price>
</Data object>

<Signature>
    <SignedInfo>
        ……..
<Reference URI="#XYZ"> Signature 1: this is signed by Signer1 with Ref    URI="agreement1"
        ……..
    </SignedInfo>
</Signature>

<Signature>
    <SignedInfo>
        ……..
<Reference URI="#XYZ"> Signature 2: this is signed by Signer1 with Ref    URI="agreement1"
        ……..
    </SignedInfo>
</Signature>
```

Figure 10 – Parallel Signature Format

3. Other stipulation like encoding and format for interoperability

No additional stipulations on encoding and format except for those listed in the ASN.1.

4. References

1. http://www.w3.org/Signature/
2. http://www.w3.org/TR/xmldsig-requirements
3. http://www.rfc-editor.org/info/rfc3275
4. The Information Technology Act, Digital Signature (End entity) Rules, 2015.

****