Cryptographic Artefact Binding Profiles

1. Introduction

The term labelling is the process of determining the appropriate metadata for a given data object, creating the metadata label and binding the metadata label to the data object. A binding is a relationship between a data object and a metadata label. A binding is realized by applying a binding mechanism. If a metadata label must be bound to a data object, both the metadata label and the data object are input to the binding mechanism. The output of the binding mechanism is the binding of a data object and metadata label (see **Figure 1**) which says that the data object and the metadata label belong together. The binding can be recorded as a structured data object, known as a Binding Data Object (BDO).

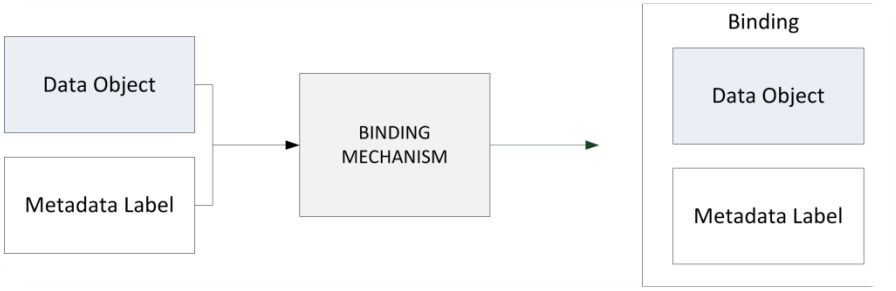

Figure 1 Creation of a binding

STANAG 4778 (Reference [2]) standardizes the binding of a data object and metadata label by specifying a common binding mechanism and a syntax for representing the BDO. However, to support information management and information sharing requirements it is necessary to further profile the application of STANAG 4778 to facilitate locating a BDO in higher level protocols, such as SMTP and HTTP, and embedding a BDO in data objects.

2. Introduction to Cryptographic Artefacts Profiles

A metadata binding provides additional information specifying which metadata belongs to which data object(s) and provides a verifiable reference between metadata and data. A non-cryptographic binding provides a reference between the metadata and the data. This reference can be structurally verified to be correct. However, no assumptions besides this can be made. In contrast, cryptographic bindings are used to provide a certain level of integrity protection, and authenticity and non-repudiation of the entity that generated the metadata binding.

A cryptographic binding (that includes cryptographic artefacts) uses cryptographic techniques and mechanisms like cryptographic digests, message authentication codes or digital signatures in order to protect the integrity of the binding. Such cryptographic techniques and mechanisms are subject to the level of assurance required for protecting the integrity of the binding and for establishing confidence for the authenticity of the entity creating the binding. The level of assurance required for protecting the integrity of the binding and for establishing confidence for the authenticity of the entity creating the binding is a matter for organizational, national or federation security policies. As such, these profiles do not mandate cryptographic techniques or mechanisms for generating a

cryptographic artefact. However, the intention is to profile the use of cryptographic protocols, which can be used to implement support for different cryptographic techniques and mechanisms, for generating cryptographic artefacts to be stored in a cryptographic binding.

The subprofiles here profile the XML Signature (XMLDSIG, Reference [3]) cryptographic protocol for generating a cryptographic artefact using digital signatures and key-hashed message authentication code (HMAC, Reference [7]) as the cryptographic techniques and mechanisms.

Table 1 below lists the supported cryptographic protocols and cryptographic mechanisms that are profiled for generating cryptographic artefacts.

Table 1: Supported Cryptographic Protocols and Mechanisms Profiles

Cryptographic Protocol	Cryptographic Mechanism	Reference
XML Signature (Reference [3])	Digital Signature	Appendix 1 Chapter 2 Appendix 1 Chapter 3
	Keyed-Hash Message Authentication Code	Appendix 1 Chapter 2 Appendix 1 Chapter 4

Further revisions to this profile may be required to add subprofiles for other cryptographic protocols such as Secure/Multipurpose Internet Mail Extensions (SMIME, Reference [8]) or JSON Web Signature (JWS, Reference [9]), for example, or to update supported cryptographic algorithms by either introducing new algorithms or deprecating existing algorithms.

3. Identification

The profile for cryptographic artefact binding is uniquely identified by the Canonical Identifier shown in Table 2.

Table 2: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:cryptoartefact
Version Identifier	urn:nato:stanag:4778:profile: cryptoartefact:1:0

It is recognized that this profile may evolve during its review cycle. For example, a review might identify:

- changes to the base standards
- support for additional algorithms
- improvements to the existing profiles based upon operational feedback

Therefore this version of the profile is uniquely identified by the Version Identifier shown in Table 2.

Subsequent versions of this profile will maintain the same Canonical Identifier, but define a new Version Identifier.

4. Standards

[1] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
[2] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium

[3] W3C XMLDSIG-CORE, 2008, "XML- Signature Syntax and Processing (Second Edition)", at http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/, W3C Recommendation, W3C, 10 June 2008
[4] Web Services Security (WS-Security), SOAP Message Security 1 (WS-Security 2004), OASIS Standard Specification, 1 February 2006
[5] W3C XPath 1.0, 1999, "XML Path Language (XPath) – Version 1.0", at http://www.w3.org/TR/xpath/, W3C Recommendation, W3C, 16 November 1999
[6] W3C XPointer, 2002, "XML Pointer Language (XPointer)", at http://www.w3.org/TR/xptr/, W3C Working Draft, W3C, 16 August 2002
[7] IETF RFC 2104, "HMAC: Keyed-Hashing for Message Authentication", at http://tools.ietf.org/html/rfc2104, February 1997
[8] IETF RFC 5751, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", at http://tools.ietf.org/html/rfc5751, January 2010
[9] IETF RFC 7515, "JSON Web Signature (JWS)", at http://tools.ietf.org/html/rfc7515, May 2015
[10] IETF RFC 6931, "Additional XML Security Uniform Resource Identifiers (URIs)", at http://tools.ietf.org/html/rfc6931, April 2013
[11] W3C XMLDSIG-2nd-Ed Errata, 2014, "Errata for XML Signature 2nd Edition", at http://www.w3.org/2008/06/xmldsigcore-errata.html, W3C Recommendation, W3C, 01 October 2014
[12] W3C XMLSEC, 2013, "XML Security Algorithm Cross-Reference", at http://www.w3.org/TR/xmlsec-algorithms, W3C Working Group Note, W3C, 11 April 2013.
[13] W3C XMLDSIG-CORE1, 2013, "XML Signature Syntax and Processing Version 1.1", at http://www.w3.org/TR/2013/REC-xmldsig-core1-20130411/, W3C Recommendation, W3C, 11 April 2013
[14] W3C XMLENC-CORE, 2002, "XML Encryption Syntax and Processing", at http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/, W3C Recommendation, W3C, 10 December 2002.
[15] W3C XMLENC-CORE1, 2013, "XML Encryption Syntax and Processing Version 1.1", at http://www.w3.org/TR/2013/REC-xmlenc-core1-20130411/, W3C Recommendation, W3C, 11 April 2013.
[16] IETF RFC 5280, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", at http://tools.ietf.org/html/rfc5280, May 2008

Appendix 1: XML Signature Cryptographic Artefact Profile

1. Introduction

XML Signature (XMLDSIG, Reference [3]) offers powerful and flexible mechanisms that can support a wide variety of cryptographic requirements. XMLDSIG provides integrity, authentication and non-repudiation services for data (including metadata) of any type. XMLDSIG is applied to arbitrary data whereby a data object is digested with the resulting value stored in an element which is then digested and cryptographically signed. XMLDSIG indicates the location of the data object either by reference (in the case of an enveloped or detached signature) or by value (in the case of an enveloping signature whereby the signature contains the data object that is to be signed).

In order to highlight the differences and avoid duplication of text from XMLDSIG, a delta specification approach has been taken. This Appendix will refer to the relevant sections of XMLDSIG and will identify any necessary clarifications and/or amendments to these sections. This approach provides traceability and puts the delta text in context. It is required that this Appendix is read together with XMLDSIG.

Figure 2 illustrates the structure of an XML Signature element including the primary sibling elements: `SignedInfo`; `SignatureValue`; `KeyInfo`; and, `Object`.

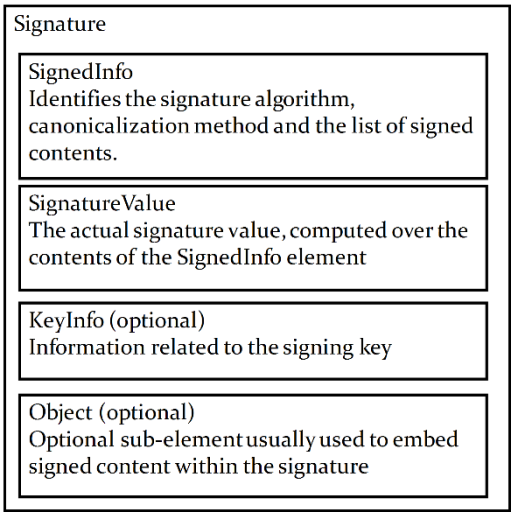

Figure 2 XML Signature Structure

This Appendix will use the same structure as illustrated in **Figure 2** to profile those requirements that are generic for XML Signature based cryptographic artefacts and to further refine those requirements for cryptographic artefacts generated with the use of digital signatures or keyed-hash message authentication codes). In particular this Appendix will be divided into the following sub sections:

- General requirements for XMLDSIG including `SignedInfo`, `SignatureValue` and `Object` elements (refer to Chapter 2);
- Specific requirements for XMLDSIG `SignedInfo` and `KeyInfo` elements related to digital signatures (refer to Chapter 4); and,

- Specific requirements for XMLDSIG `SignedInfo` and `KeyInfo` elements related to keyed-hashed message authentication codes (refer to Chapter 3).

Example Binding Data Objects containing cryptographic artefacts conformant with this profile are illustrated in Chapter 5.

The notational conventions used for this Appendix are as follows:

- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119, 1997].
- Words in *italics* indicate terms derived from Reference [2].
- `Courier font` indicates syntax derived from various W3C XML Signature (Reference [3]) standard referenced in this Appendix.
- *`Courier font`* indicates syntax derived from Web Services Security (WSS) (Reference [4]) standard Section 10 referenced in this Appendix.

2. General XMLDSIG Requirements

Unless otherwise stated, all statements that apply to XMLDSIG also apply to this profile.

An entity that creates XML Signatures conformant with this profile (known as Originator) is REQUIRED to perform the processing rules for Core Generation as specified in XMLDSIG Section 3.1.

An entity that interprets and processes XML Signatures conformant with this profile (known as Recipient) is REQUIRED to perform the processing rules for Core Validation as specified in XMLDSIG Section 3.2.

Signature Types

Three types of signatures exist in XMLDSIG: enveloping signatures whereby the signature envelopes the data object to be signed; enveloped signatures whereby the signature is embedded within the data object; and, detached signatures whereby the signature and the data object reside independently.

Enveloping, Enveloped and Detached signature types are supported in this profile.

Same-Document URI-References

This section refers to XMLDSIG Section 4.3.3.1, 4.3.3.2 and 4.3.3.3.

The significance of the URI fragment identifier for dereferencing subsets of data objects is a function of the type (media type) of the data object. Identification for the media type of a data object is supported in the general binding mechanism with the use of the *xmime:contentType* attribute. The *xmime:contentType* attribute for non-XML is a required attribute of the *DataReference* and *MetadataReference* elements.

In the case where the *xmime:contentType* attribute is present in the *DataReference* or *MetadataReference* element, the *xmime:contentType* attribute value specifies a non-XML data object type and the URI attribute value of the *DataReference* or *MetadataReference* element is deemed to be a '`same-document`' reference (as specified in XMLDSIG Section 4.3.3.2) the following requirements are to be followed:

- Originator MUST create a `Manifest` element for each *DataReference* or *MetadataReference* elements contained in the *bindingInformation* that includes a `Reference` element (as specified in Manifest section of Chapter 2);
- The `Manifest` element that the Originator creates MUST be stored as a child element of an `Object` element;
- Recipient SHOULD perform the following additional Core Validation processing rules:
    - For each `Reference` in the `Manifest`:
        - Obtain the data object to be digested located by the `URI` attribute in the `Reference` element (According to the semantics specified for the URI fragment identifier defined by the media type);
        - Digest the resulting data object using the `DigestMethod` (as specified in the Reference section in Chapter 2).
        - Compare the generated digest value against `DigestValue` in the `Manifest Reference`; if there is any mismatch, validation fails.

XML Security Uniform Resource Identifiers (URIs)

XML security algorithm identifiers have been defined in a number of different specifications such as XML Signature, XML Encryption and RFCs. XML Security Algorithm Cross-Reference (Reference [12]) provides a non-normative list of identifiers that have been defined by XML Signature (References [3] and [13]), XML Encryption (References [14] and [15]) and Additional XML Security Uniform Resource Identifiers (URIs, Reference [10]).

This Appendix profiles the use of those algorithm identifiers listed in Reference [12] specifying whether support for that algorithm is mandatory, optional or prohibited for signature generation.

Mandatory and optional algorithms on signature generation MUST be supported on signature validation.

Prohibited algorithms on signature generation MAY be supported on signature validation.

Core Signature Syntax

This section refers to XMLDSIG Section 4.

Signature

This section refers to XMLDSIG Section 4.1.

In the case where a cryptographic binding is required the *bindingInformation* element (specified in Reference [2]) MUST contain at least one *Signature* element.

SignatureValue

This section refers to XMLDSIG Section 4.2.

SignedInfo

This section refers to XMLDSIG Section 4.3.

CanonicalizationMethod

This section refers to XMLDSIG Section 4.3.1.

The `CanonicalizationMethod Algorithm` attribute MUST be one of the following:

- http://www.w3.org/TR/2001/REC-xml-c14n-20010315
- http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments
- http://www.w3.org/2006/12/xml-c14n11
- http://www.w3.org/2006/12/xml-c14n11#WithComments
- http://www.w3.org/2001/10/xml-exc-c14n#
- http://www.w3.org/2001/10/xml-exc-c14n#WithComments
- http://www.w3.org/2010/10/xml-c14n2.

SignatureMethod

This section refers to XMLDSIG Section 4.3.2.

The `SignatureMethod Algorithm` attribute is REQUIRED.

The value of the SignatureMethod Algorithm is further specified depending on the cryptographic technique and mechanism being used (refer to Chapter 3 for Digital Signatures or Chapter 4 for HMAC).

Reference

This section refers to XMLDSIG Section 4.3.3.

For each *DataReference* or *MetadataReference* element included in a *bindingInformation* element there MUST be a Reference element

In the use case identified in Same-Document URI-References there MUST be a Reference element that identifies the Manifest element.

For each *MetadataBinding* element included in the *bindingInformation* element there MUST be a Reference element that identifies each *MetadataBinding* element.

URI

This section refers to XMLDSIG Section 4.3.3.1.

For each *DataReference* or *MetadataReference* element included in a *bindingInformation* element that contains a *URI* attribute with a value there MUST be a Reference element with the same URI attribute value, except in the case identified in Same-Document URI-References.

In the case identified in Same-Document URI-References there MUST be a URI attribute present with the value referencing the *Manifest* element.

For each *MetadataBinding* element included in the *bindingInformation* element there MUST be a Reference URI attribute with a shortname XPointer (Reference [6]) as the attribute value that identifies each *MetadataBinding* element.

Transforms

This section refers to XMLDSIG Section 4.3.3.4.

For Embedded BDOs in an XML data object an Enveloped Binding Data Object transform MUST first be applied to remove the *BindingInformation* element from the digest calculation of the Reference element containing the *BindingInformation* element. The Enveloped Binding Data Object transform element MUST have Transform Algorithm attribute value of *http://www.w3.org/TR/1999/REC-xpath-19991116* and MUST contain the following XPath element:

```
<XPath>
   not(ancestor-or-self::*[local-name() = 'BindingInformation' and
   namespace-uri() = 'urn:nato:stanag:4778:bindinginformation:1:0'
</XPath>
```

For Embedded BDOs where the *xmime:contentType* attribute is present in the *DataReference* element and the *xmime:contentType* attribute value specifies a non-XML data object type the use of the Enveloped Binding Data Object does not apply. In this use case the signature generation and signature validation process SHALL first exclude the Embedded Binding Data Object (the *BindingInformation* element) from the digest calculation of the Reference element containing the *BindingInformation* element.

For each *DataReference* or *MetadataReference* element included in a *bindingInformation* element that contains a *Transforms* element the first (or next in the case of Embedded BDOs) `Transform` element of the `Reference Transforms` element MUST be the *Transform* element from the *DataReference* or *MetadataReference* element.

For each *MetadataBinding* element included in the *bindingInformation* element there MAY be a `Transform` element (child of the `Transforms` element) that includes an XPath (Reference [5]) expression to identify *MetadataBinding* element.

For each *MetadataBinding*, *DataReference*, and *MetadataReference* that is identified by an XPath expression the `Transform` element MUST have an `Algorithm` attribute with the value '*http://www.w3.org/TR/1999/REC-xpath-19991116*'.

Other `Transform` elements MAY be present.

For other `Transform` elements the `Transform Algorithm` attribute MUST have one of the following values:

- http://www.w3.org/2000/09/xmldsig#base64
- http://www.w3.org/TR/1999/REC-xpath-19991116
- http://www.w3.org/2002/06/xmldsig-filter2
- http://www.w3.org/2000/09/xmldsig#enveloped-signature
- http://www.w3.org/TR/1999/REC-xslt-19991116
- http://www.w3.org/TR/2001/REC-xml-c14n-20010315
- http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments
- http://www.w3.org/2006/12/xml-c14n11
- http://www.w3.org/2006/12/xml-c14n11#WithComments
- http://www.w3.org/2001/10/xml-exc-c14n#
- http://www.w3.org/2001/10/xml-exc-c14n#WithComments
- http://www.w3.org/2010/10/xml-c14n2.

DigestMethod

This section refers to XMLDSIG Section 4.3.3.5.

The `DigestMethod Algorithm` attribute MUST conform to the specifications detailed in **Table 3**.

Table 3: DigestMethod Algorithm Identifiers

Algorithm Identifier	Mandatory/Optional/ Prohibited
http://www.w3.org/2001/04/xmldsig-more#md5	Prohibited
http://www.w3.org/2000/09/xmldsig#sha1	Prohibited
http://www.w3.org/2001/04/xmldsig-more#sha224	Optional
http://www.w3.org/2001/04/xmlenc#sha256	Mandatory
http://www.w3.org/2001/04/xmldsig-more#sha384	Optional
http://www.w3.org/2001/04/xmlenc#sha512	Optional
http://www.w3.org/2001/04/xmlenc#ripemd160	Optional

DigestValue

This section refers to XMLDSIG Section 4.3.3.6.

KeyInfo

This section refers to XMLDSIG Section 4.4.

The `KeyInfo` element is REQUIRED.

Refer to the relevant section, dependent upon the cryptographic technique and mechanism being used (refer to Chapter 3 for Digital Signatures or Chapter 4 for HMAC), for further profiling of the `KeyInfo` element.

Object

This section refers to XMLDSIG Section 4.5.

The `Object` element is REQUIRED.

Additional Signature Syntax

This section refers to XMLDSIG Section 5.

Manifest

This section refers to XMLDSIG Section 5.1.

The `Manifest` element is REQUIRED only to support the use case for Same-Document URI-References.

The Originator MUST obtain the data object to be digested by dereferencing the *URI* attribute value in the *MetadataReference* or *DataReference* element in accordance to the semantics specified for the URI fragment identifier defined by the media type (identified in the *MetadataReference contentType* or *DataReference contentType* attribute value).

The Originator MUST perform the processing rules for Reference Generation as specified in XMLDSIG Section 3.1.1 with the following constraint:

The `Reference` element `URI` attribute value MUST be the same value as the *DataReference* (or *MetadataReference*) *URI* attribute value.

In other cases the use of the `Manifest` element is NOT REQUIRED.

In the case where the use of the `Manifest` element is required it is RECOMMENDED that the originator create a `Reference` element, including the identification of the `Manifest` element, any transform elements, the digest algorithm and the `DigestValue` in order to be included in the signature

SignatureProperties

This section refers to XMLDSIG Section 5.2.

TimeStamp

This section refers to Web Services Security (WSS) (Reference [4]) Section 10.

The *TimeStamp* element MUST be present indicating the time that the cryptographic binding was created as a value of the *Created* element.

The *ValueType* attribute of the *Created* element MUST be *xsd:dateTime*.

The *Expires* element (child element of the *TimeStamp* element) is NOT REQUIRED.

The inclusion of an indication when the cryptographic binding was created supports the following two use cases:

1) Detection of replay attacks; and,
2) A valid cryptographic binding at time of signing, however, the key material used for creating the signature may have expired, been revoked or other.

It is RECOMMENDED that the originator create a `Reference` element, including the identification of the *TimeStamp* element in order to be included in the signature.

3. Digital Signature Cryptographic Artefact

Implementations that use digital signatures as the cryptographic mechanism for producing cryptographic artefacts are REQUIRED to be conformant with Chapter 2 and this Chapter.

SignedInfo

This section refers to XMLDSIG Section 4.3.

SignatureMethod

This section refers to XMLDSIG Section 4.3.2.

The `SignatureMethod Algorithm` attribute MUST conform to the specifications detailed in **Table 4**.

Table 4: SignatureMethod (PKI) Algorithm Identifiers

Algorithm Identifier	Mandatory/Optional/Prohibited
http://www.w3.org/2000/09/xmldsig#dsa-sha1	Prohibited
http://www.w3.org/2009/xmldsig11#dsa-sha256	Optional
http://www.w3.org/2001/04/xmldsig-more#rsa-md5	Prohibited
http://www.w3.org/2000/09/xmldsig#rsa-sha1	Prohibited
http://www.w3.org/2001/04/xmldsig-more#rsa-sha224	Optional
http://www.w3.org/2001/04/xmldsig-more#rsa-sha256	Mandatory
http://www.w3.org/2001/04/xmldsig-more#rsa-sha384	Optional
http://www.w3.org/2001/04/xmldsig-more#rsa-sha512	Optional
http://www.w3.org/2001/04/xmldsig-more#rsa-ripemd160	Optional
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1	Prohibited
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha224	Optional
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256	Mandatory
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha384	Optional
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512	Optional

KeyInfo

This section refers to XMLDSIG Section 4.4.

The `KeyInfo` element is REQUIRED.

KeyName

This section refers to XMLDSIG Section 4.4.1.

The `KeyName` element SHALL NOT be present.

KeyValue

This section refers to XMLDSIG Section 4.4.2.

The `KeyValue` MAY be present.

RetrievalMethod

This section refers to XMLDSIG Section 4.4.3.

The `RetrievalMethod` SHALL NOT be present.

X509Data

This section refers to XMLDSIG Section 4.4.4.

The `X509Data` element is REQUIRED.

In strategic systems with high throughput, certificates MUST be included.

X.509 version 3 certificates MUST be supported.

The certificate profile specified in Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (Reference [16]) MUST be supported.

The Originator SHOULD include at least one chain of certificates up to, but not including, a Certificate Authority (CA) that it believes that the Recipient may trust as authoritative.

Each certificate MUST be included in an `X509Certificate` element.

The Recipient SHOULD be able to handle an arbitrarily large number of certificates and chains.

In those cases where certificates may not be transmitted one of the `X509IssuerSerial`, `X509SKI` and `X509SubjectName` elements MUST be present.

The `X509CRL` element is NOT REQUIRED.

The CRL SHOULD be looked up based on the CRL Distribution Point (CDP) contained in the certificate.

The CRL profile specified in Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (Reference [16]) MUST be supported.

PGPData

This section refers to XMLDSIG Section 4.4.5.

The `PGPData` element SHALL NOT be present.

SPKIData

This section refers to XMLDSIG Section 4.4.6.

The `SPKIData` element SHALL NOT be present.

MgmtData

This section refers to XMLDSIG Section 4.4.7.

The `MgmtData` element SHALL NOT be present.

4. Keyed-Hash Message Authentication Code Cryptographic Artefact

Implementations that use keyed-hash message authentication codes (Reference [7]) as the cryptographic mechanism for producing cryptographic artefacts are REQUIRED to be conformant with Chapter 2 and this Chapter.

SignedInfo

This section refers to XMLDSIG Section 4.3.

SignatureMethod

This section refers to XMLDSIG Section 4.3.2.

The `SignatureMethod Algorithm` attribute MUST conform to the specifications detailed in **Table 5**.

Table 5: SignatureMethod (HMAC) Algorithm Identifiers

Algorithm Identifier	Mandatory/Optional/Prohibited
http://www.w3.org/2000/09/xmldsig#hmac-sha1	Prohibited
http://www.w3.org/2001/04/xmldsig-more#hmac-sha224	Optional
http://www.w3.org/2001/04/xmldsig-more#hmac-sha256	Mandatory
http://www.w3.org/2001/04/xmldsig-more#hmac-sha384	Optional
http://www.w3.org/2001/04/xmldsig-more#hmac-sha512	Optional
http://www.w3.org/2001/04/xmldsig-more#hmac-ripemd160	Optional

In the case whereby the `HMACOutputLength` is used for HMAC algorithms the errata to XMLDSIG (Reference [11]) MUST be followed.

KeyInfo

This section refers to XMLDSIG Section 4.4.

The `KeyInfo` element is REQUIRED.

KeyName

This section refers to XMLDSIG Section 4.4.1.

The `KeyName` element MAY be present.

KeyValue

This section refers to XMLDSIG Section 4.4.2.

The `KeyValue` SHALL NOT be present.

RetrievalMethod

This section refers to XMLDSIG Section 4.4.3.

The `RetrievalMethod` SHALL NOT be present.

X509Data

This section refers to XMLDSIG Section 4.4.4.

The `X509Data` SHALL NOT be present.

PGPData

This section refers to XMLDSIG Section 4.4.5.

The `PGPData` element SHALL NOT be present.

SPKIData

This section refers to XMLDSIG Section 4.4.6.

The `SPKIData` element SHALL NOT be present.

MgmtData

This section refers to XMLDSIG Section 4.4.7.

The `MgmtData` element SHALL NOT be present.

5. Examples

This Chapter contains fictitious examples that illustrate cryptographic Binding Data Objects (BDOs) that contain cryptographic artefacts conformant with this appendix. All examples given in this appendix use Confidentiality Metadata Labels (Reference [1]) as example metadata.

The examples are provided as self-explanatory representations of BDOs.

```
<mb:BindingInformation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
  <Signature Id="id-a99fac99-513d-4b08-8158-ef862e4d9f80"
xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256"
/>
      <Reference URI="#id-66bb29e1-9696-4ea0-be3c-f7d0096a0d81">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>9JBAVs2gUWUzFh8uUl1ubXW13VgQxli3NM+CF0vQG14=</DigestValue>
      </Reference>
      <Reference URI="#id-d55d0123-babc-467f-b309-62e95291a9e4">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>8G8AHBPiAJ+W6PUOq+W/Vua+iO7Zj6GzooPRmkqtqnY=</DigestValue>
      </Reference>
      <Reference URI="#id-b3eaf318-700f-4740-b43e-2def8d98db81">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>Kx02/WnFE/2MN7lEuWemAiDetsJZ+8lJt4nvg4GyRNc=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>g3nzbBiu7msmVHfCjmVqqSiimlASoBSM/hxqFN7YxH0=</SignatureValue>
    <KeyInfo Id="id-b3eaf318-700f-4740-b43e-2def8d98db81">
      <KeyName>HMAC_SECRET_KEY</KeyName>
    </KeyInfo>
    <Object Id="id-17250b2d-f0f5-4457-9e21-23db31e3460d">
      <SignatureProperties Id="id-d55d0123-babc-467f-b309-62e95291a9e4">
        <SignatureProperty>
          <wsu:TimeStamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
            <wsu:Created>2015-11-13T15:58:44Z</wsu:Created>
          </wsu:TimeStamp>
        </SignatureProperty>
      </SignatureProperties>
    </Object>
  </Signature>
  <mb:MetadataBindingContainer>
    <mb:MetadataBinding Id="#id-66bb29e1-9696-4ea0-be3c-f7d0096a0d81">
      <mb:Metadata>
        <slab:originatorConfidentialityLabel
xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
          <slab:ConfidentialityInformation>
            <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
            <slab:Classification>UNCLASSIFIED</slab:Classification>
          </slab:ConfidentialityInformation>
          <slab:CreationDateTime>
         2015-09-30T12:30:00Z
        </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
      </mb:Metadata>
      <mb:Data>
        <Document xmlns="">
          <Title>BDO Examples</Title>
          <Author>alan.ross@reach.nato.int</Author>
          <Abstract>
```

```
            Example XML File to support illustration of different types of BDO and
cryptographic artefacts
          </Abstract>
          <Introduction>....</Introduction>
          <Chapter Id="chapter-1">
            <Paragraph Id="para-1-1" />
            <Paragraph Id="para-1-2" />
          </Chapter>
          <Chapter Id="chapter-2">
            <Paragraph Id="para-2-1" />
            <Paragraph Id="para-2-2" />
          </Chapter>
        </Document>
      </mb:Data>
    </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
</mb:BindingInformation>
```

Figure 3 Encapsulating Cryptographic BDO Containing an Enveloped Signature with a Keyed-Hash Message Authentication Code Cryptographic Artefact

```
<mb:BindingInformation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
  <Signature Id="id-fb00da79-4b32-4fcc-a302-4dbf789212e3"
xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <Reference URI="#id-20c07ca8-6960-4a36-bdd1-e3cb299f82c3">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>fAXcjRa4zlLyB+lchyBK/9JzlsoZSbxNCmr/27nA9aI=</DigestValue>
      </Reference>
      <Reference URI="#id-82744679-a547-40aa-a683-cf97619054fe">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>j5AgAamc6cv54VDzl0kDlQ4wYZLLAU3761eFOUWvtX0=</DigestValue>
      </Reference>
      <Reference URI="#id-9920a48c-c3a1-45d0-a81c-1ce04d1d8de6">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>hWUoi0gFxnFsGnHJO/V2eNg/silda814PSP2/WlsqtU=</DigestValue>
      </Reference>
    </SignedInfo>

<SignatureValue>gItAuwdEykw5xDht50TOei1xfT0q7KLaUXm4w/2rnpTjoxiODTI3Wr8D4fmx/404bVrX23S
tY6HHT/dxDPcgODa+K9YL/pl3y8RvIrfWGhiZReY5AUj1EF3mxI22ari/ao0shKe18aPJ0J2RmGH3t30qrHfvUX
cIcREIOT1S6GajpNCOJPYoa9yb400MOx0oRHXkFegnQ5eXeSBIh2u4DhwL0I4GSeuYA9FVt8qyvla9EnTTS6fG2
+gLjd6YEQzfIBvVtrY5b9WnhqqiHy5tyepZgVtMSEXrukWrNELpvwC467KR+MincgUA9RlsAEvCBaR4oQKTUOxB
Q5tD+N/FzQ==</SignatureValue>
    <KeyInfo Id="id-9920a48c-c3a1-45d0-a81c-1ce04d1d8de6">
      <X509Data>
              <X509Certificate>MIIDM……wIBAgIJAI29/+A/MN7RPAx5eOKQg==</X509Certificate>
      </X509Data>
    </KeyInfo>
    <Object Id="id-63fc02c0-10b6-49fd-9759-7bfb1d52ecf7">
      <SignatureProperties Id="id-82744679-a547-40aa-a683-cf97619054fe">
        <SignatureProperty>
          <wsu:TimeStamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
            <wsu:Created>2015-11-13T16:01:38Z</wsu:Created>
          </wsu:TimeStamp>
        </SignatureProperty>
      </SignatureProperties>
    </Object>
  </Signature>
```

```
    <mb:MetadataBindingContainer>
      <mb:MetadataBinding Id="#id-20c07ca8-6960-4a36-bdd1-e3cb299f82c3">
        <mb:Metadata>
          <slab:originatorConfidentialityLabel
xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
            <slab:ConfidentialityInformation>
              <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
              <slab:Classification>UNCLASSIFIED</slab:Classification>
            </slab:ConfidentialityInformation>
            <slab:CreationDateTime>
             2015-09-30T12:30:00Z
          </slab:CreationDateTime>
          </slab:originatorConfidentialityLabel>
        </mb:Metadata>
        <mb:Data>
          <Document xmlns="">
            <Title>BDO Examples</Title>
            <Author>alan.ross@reach.nato.int</Author>
            <Abstract>
             Example XML File to support illustration of different types of BDO and
cryptographic artefacts
            </Abstract>
            <Introduction>....</Introduction>
            <Chapter Id="chapter-1">
              <Paragraph Id="para-1-1" />
              <Paragraph Id="para-1-2" />
            </Chapter>
            <Chapter Id="chapter-2">
              <Paragraph Id="para-2-1" />
              <Paragraph Id="para-2-2" />
            </Chapter>
          </Document>
        </mb:Data>
      </mb:MetadataBinding>
    </mb:MetadataBindingContainer>
  </mb:BindingInformation>
```

Figure 4 Encapsulating Cryptographic BDO Containing an Enveloped Signature with a Digital Signature Cryptographic Artefact

```
  <Document xmlns="http://example.com/doc">
    <Title>BDO Examples</Title>
    <Author>alan.ross@reach.nato.int</Author>
    <Abstract>
    Example XML File to support illustration of different types of BDO and cryptographic
artefacts
    </Abstract>
    <Introduction>....</Introduction>
    <Chapter Id="chapter-1">
      <Paragraph Id="para-1-1" />
      <Paragraph Id="para-1-2" />
    </Chapter>
    <Chapter Id="chapter-2">
      <Paragraph Id="para-2-1" />
      <Paragraph Id="para-2-2" />
    </Chapter>
    <mb:BindingInformation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
      <Signature Id="id-134ce280-1682-4963-b868-6621b480ce26"
xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256"
/>
          <Reference URI="">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
                <XPath>not(ancestor-or-self::*[local-name() = 'BindingInformation' and
namespace-uri() = 'http://www.nato.int/2014/06/nl/mb'])</XPath>
              </Transform>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
```

```
            <DigestValue>RYxJZ8BN/MR2D0BDxiCxGSDaQvGFKQ86udb0Ov5A2s4=</DigestValue>
          </Reference>
          <Reference URI="#id-c9e4f1c8-ad34-4d4d-9909-827570de41a2">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
            <DigestValue>WKOWdda84YLuSqbaZsS8LQ6kqF6HR0dfC+iz/e+KPf0=</DigestValue>
          </Reference>
          <Reference URI="#id-1bd95780-277f-44b3-99fd-b2b69505ae5a">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
            <DigestValue>UbMTebL9lKFARnG1qWOpQ1DiuCFPzs6W1hse9gPOxUk=</DigestValue>
          </Reference>
          <Reference URI="#id-001c1a07-74c4-4815-aecd-dd1bcba8bc9c">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
            <DigestValue>z7+6QZiSTqYMHCiy9o3uxGfA8q5ScEeHlHZs3w9+8S4=</DigestValue>
          </Reference>
        </SignedInfo>
        <SignatureValue>dk7Ds4Atik6yF/wKZjOIDVGGyv1rigTDLj6gRsqCTHY=</SignatureValue>
        <KeyInfo Id="id-001c1a07-74c4-4815-aecd-dd1bcba8bc9c">
          <KeyName>HMAC_SECRET_KEY</KeyName>
        </KeyInfo>
        <Object Id="id-4dcc6c48-6ed0-4cf0-b386-b85f7ee0c826">
          <SignatureProperties Id="id-1bd95780-277f-44b3-99fd-b2b69505ae5a">
            <SignatureProperty>
              <wsu:TimeStamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
    200401-wss-wssecurity-utility-1.0.xsd">
                <wsu:Created>2015-11-13T16:07:37Z</wsu:Created>
              </wsu:TimeStamp>
            </SignatureProperty>
          </SignatureProperties>
        </Object>
      </Signature>
      <mb:MetadataBindingContainer>
        <mb:MetadataBinding Id="#id-c9e4f1c8-ad34-4d4d-9909-827570de41a2">
          <mb:Metadata>
            <slab:originatorConfidentialityLabel
    xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
              <slab:ConfidentialityInformation>
                <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
                <slab:Classification>UNCLASSIFIED</slab:Classification>
              </slab:ConfidentialityInformation>
              <slab:CreationDateTime>
         2015-09-30T12:30:00Z
        </slab:CreationDateTime>
            </slab:originatorConfidentialityLabel>
            <slab:alternateConfidentialityLabel
    xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
              <slab:ConfidentialityInformation>
                <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
                <slab:Classification>UNCLASSIFIED</slab:Classification>
              </slab:ConfidentialityInformation>
              <slab:CreationDateTime>
         2015-09-30T12:30:00Z
        </slab:CreationDateTime>
            </slab:alternateConfidentialityLabel>
          </mb:Metadata>
          <mb:DataReference URI="" />
        </mb:MetadataBinding>
      </mb:MetadataBindingContainer>
    </mb:BindingInformation>
  </Document>
```

Figure 5 Embedded Cryptographic BDO Containing an Enveloped Signature with a Keyed-Hash Message Authentication Code Cryptographic Artefact

```
<Document xmlns="http://example.com/doc">
  <Title>BDO Examples</Title>
  <Author>alan.ross@reach.nato.int</Author>
  <Abstract>
  Example XML File to support illustration of different types of BDO and cryptographic
artefacts
  </Abstract>
  <Introduction>....</Introduction>
  <Chapter Id="chapter-1">
    <Paragraph Id="para-1-1" />
    <Paragraph Id="para-1-2" />
  </Chapter>
  <Chapter Id="chapter-2">
    <Paragraph Id="para-2-1" />
    <Paragraph Id="para-2-2" />
  </Chapter>
  <mb:BindingInformation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
    <Signature Id="id-3a7079e1-adeb-47b0-a4df-86a5f2962f57"
xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"
/>
        <Reference URI="#para-2-2">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>0JsT5SNKuCYoe91tl8n590Hcy/UivrId3Zf6kJy7pdg=</DigestValue>
        </Reference>
        <Reference URI="#id-b073db91-a8b3-4905-809d-82e92b0d0ecc">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>a3yUgG8j0eIPI6ZSw7aw4JPHO1SBglS0+Fb7lwVmMeo=</DigestValue>
        </Reference>
        <Reference URI="#id-7d5d0648-59c1-48a9-a3bc-a07a24f0a67b">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>zMHgHTwG+OtqPY8+T4cwYGby2UoSv71QJ2eU0peB5ds=</DigestValue>
        </Reference>
        <Reference URI="#id-45f67abd-5803-4933-acb8-5061adde54f4">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>g8jESHIgXr4bGZFwOzh2O4r8Vv0y6jfH7qKgQTGV9ww=</DigestValue>
        </Reference>
      </SignedInfo>

<SignatureValue>C1yPwzpU/ngO42sXo2HHZTtbXNTe2FAXf2RivMy5u6z/xoNlmi/mHm5ejZPFWkoGaUmWDad
REcc5lI6XBYXeks2YVyMh05uDRCQLPYNkIAx3BpUFH7y9JUklj4WvlDBeZ2GwNhp463QMvn8pF35cXw1f86Vc0M
3CtAm5MNbnS6BqqswdygCF/HivjHcQSnYGRhI4vegelwfYyhFRHQQ1OE3ytUDR8VLKZfgyK3M6mcQjvlHtL2qjR
xMHrkQQtt8oBQk6iAWxYgbqeIzqw3cIYL5jb/ML2UOycGgwUIqGFx95EouKuOMZSN8e2dnaVaHp26XlzpdJkyTk
Vr5/T7v3hA==</SignatureValue>
      <KeyInfo Id="id-45f67abd-5803-4933-acb8-5061adde54f4">
        <X509Data>
              <X509Certificate> MIIDM……wIBAgIJAI29/+A/MN7RPAx5eOKQg==</X509Certificate>
        </X509Data>
      </KeyInfo>
      <Object Id="id-221fefa8-fd81-4f98-8784-ac4a08e4eece">
        <SignatureProperties Id="id-7d5d0648-59c1-48a9-a3bc-a07a24f0a67b">
          <SignatureProperty>
            <wsu:TimeStamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
              <wsu:Created>2015-11-13T16:04:59Z</wsu:Created>
            </wsu:TimeStamp>
          </SignatureProperty>
        </SignatureProperties>
      </Object>
```

```
        </Signature>
        <mb:MetadataBindingContainer>
          <mb:MetadataBinding Id="#id-b073db91-a8b3-4905-809d-82e92b0d0ecc">
            <mb:Metadata>
              <slab:originatorConfidentialityLabel
    xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
                <slab:ConfidentialityInformation>
                  <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
                  <slab:Classification>UNCLASSIFIED</slab:Classification>
                </slab:ConfidentialityInformation>
                <slab:CreationDateTime>
           2015-09-30T12:30:00Z
          </slab:CreationDateTime>
              </slab:originatorConfidentialityLabel>
            </mb:Metadata>
            <mb:DataReference URI="" />
          </mb:MetadataBinding>
          <mb:MetadataBinding>
            <mb:Metadata>
              <slab:originatorConfidentialityLabel
    xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
                <slab:ConfidentialityInformation>
                  <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
                  <slab:Classification>CONFIDENTIAL</slab:Classification>
                </slab:ConfidentialityInformation>
                <slab:CreationDateTime>
           2015-09-30T12:30:00Z
          </slab:CreationDateTime>
              </slab:originatorConfidentialityLabel>
            </mb:Metadata>
            <mb:DataReference URI="#para-2-1" />
          </mb:MetadataBinding>
        </mb:MetadataBindingContainer>
      </mb:BindingInformation>
    </Document>
```

Figure 6 Embedded Cryptographic BDO Containing a Detached Signature with a Digital Signature Cryptographic Artefact