



NATO Communications and Information Agency
Agence OTAN d'information et de communication

TN-1491 Edition 2

PROFILES FOR BINDING METADATA TO A DATA OBJECT

Alan Murdock, Graeme Lunt, Alan Ross



CONDITIONS OF RELEASE

With reference to NATO Documents C-M(2002)49 and AC/322-D/1, this document is released to a NATO Government at the direction of the NATO Communications and Information (NCI) Agency subject to the following conditions:

1. The recipient NATO Government agrees to use its best endeavours to ensure that the information herein disclosed, whether or not it bears a security classification, is not dealt with in any manner (a) contrary to the intent of the provisions of the Charter of the NATO Communications and Information Organization, or (b) prejudicial to the rights of the owner thereof to obtain patent, copyright or other likely statutory protection therefor.
2. If the technical information was originally released to the Agency by a NATO Government subject to restrictions clearly marked on this document the recipient NATO Government agrees to abide by the terms of the restrictions so imposed by the releasing Government.

PROFILES FOR BINDING METADATA TO A DATA OBJECT

Alan Murdock, Graeme Lunt, Alan Ross

Keywords: STANAG 4778, Metadata Binding Profiles, NISP

Abstract

This document describes the application of the STANAG 4778 Metadata Binding Mechanism to specific data formats and protocols by defining a suite of Binding Profiles. These distinct binding profiles are specified for the following protocols and data formats:

- *Web Services (SOAP-based and REST-based web services);*
- *SMTP/MIME internet email messaging;*
- *Common XML Artefacts (e.g. XML Schemas, Stylesheets);*
- *Collaboration (Text-based instant messaging);*
- *Document management (including Office Tools)*
- *Extensible Metadata Platform (XMP);*
- *Web Service Messaging Profile (WSMP); and*
- *Arbitrary Files.*

In addition a Cryptographic Artefact Binding Profile is defined to support strong binding of metadata to data objects.

These profiles represent the first suite of Binding Profiles and will be extended through further editions of this document.

The work described in this report was carried out under Project SPW011855 within ACT Scientific Program of Work 2017 and under Project NCB011722 within the C3S Program of Work 2017 and was concluded in September 2017.

Approved: _____

NATO Communications and Information Agency
The Hague

This document consists
of iii + 95 pages
(excluding covers)

This page is left blank intentionally

SUMMARY

STANAG 4778 - Metadata Binding Mechanism specifies a method for binding metadata information (including confidentiality metadata labels) to finite data objects. As this STANAG and companion STANAGs, STANAG 5636 - NATO Core Metadata Specification (NCMS) and STANAG 4774 – Confidentiality Metadata Label Syntax, progress towards ratification, there is an urgent and significant need for complementary Binding Profiles to reduce identified risks to capability procurement for common funded programmes in the NATO Enterprise. STANAG 4774 (when ratified) cannot be mandated or implemented without a complementary binding mechanism (STANAG 4778) supplemented by Binding Profiles for specific protocols and data types.

This document captures a number of Binding Profiles that use the mechanism defined in STANAG 4778 to allow the binding of the metadata to a selected data object.

These Binding Profiles have been under continual validation since the XML Labelling Guard deployment to the NATO missions in 2011. This continued during CWIX where successful validation efforts have been executed using newly defined profiles.

The Binding Profiles specified in Edition 2 of this document have been submitted to the IP CaT for inclusion in STANAG 5524 – NATO Interoperability Standards and Profiles (NISP) Version 11. Additional Binding Profiles may be developed and supplement those in Edition 2. This will result in further editions of this document to be issued on a regular basis.

This page is left blank intentionally

TABLE OF CONTENTS

	Page
STANAG 4778 BINDING PROFILES	1
ANNEX A: CRYPTOGRAPHIC ARTEFACT BINDING PROFILES	1
ANNEX B: SIMPLE MAIL TRANSFER PROTOCOL BINDING PROFILE	B-1
ANNEX C: EXTENSIBLE MESSAGE AND PRESENCE PROTOCOL BINDING PROFILE	C-1
ANNEX D: OFFICE OPEN XML FORMATS BINDING PROFILE	D-1
ANNEX E: SIMPLE OBJECT ACCESS PROTOCOL BINDING PROFILE	E-1
ANNEX F: REPRESENTATIONAL STATE TRANSFER BINDING PROFILE	F-1
ANNEX G: GENERIC OPEN PACKAGING CONVENTION BINDING PROFILE	G-1
ANNEX H: SIDECAR FILES BINDING PROFILE	H-1
ANNEX I: EXTENSIBLE METADATA PLATFORM BINDING PROFILE	I-1
ANNEX J: WEB SERVICE MESSAGING PROFILE BINDING PROFILE	J-1
ANNEX K: COMMON XML ARTEFACTS BINDING PROFILE	K-1

This page is left blank intentionally

STANAG 4778 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 the data object(s) and the metadata label(s). 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).

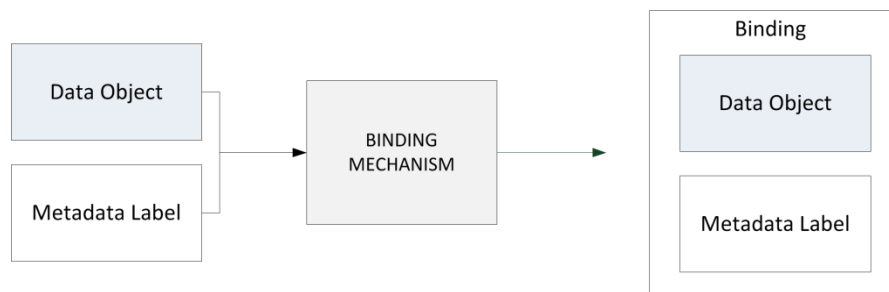


Figure 1: Creation of a binding

STANAG 4778 (Reference [3]) 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.

This document describes the application of the STANAG 4778 Metadata Binding Mechanism to specific data formats and protocols. It provides distinct binding profiles for the following protocols and data formats:

- Web Services (SOAP-based and REST-based web services);
- SMTP/MIME internet email messaging;
- Common XML Artefacts (e.g. XML schemas, stylesheets);
- Collaboration (Text-based instant messaging);
- Document management (including Office Tools);
- Extensible Metadata Platform (XMP); and
- Arbitrary Files.

Additionally, distinct profiles are provided to guide the application of strong bindings to any of the protocols and data formats indicated. A strong binding uses cryptographic techniques and mechanisms such as cryptographic digests, message authentication codes or digital signatures in order to protect the binding. Two distinct cryptographic bindings are provided:

- XML Signature cryptographic protocol using digital signatures; and

- XML Signature cryptographic protocol using Key-Hashed Message Authentication Code (HMAC).

This list of Binding Profiles is not exhaustive and new profiles may be added through the NISP RFCP process. In addition, it is quite possible that more than one Binding Profile will be defined for a particular protocol or data format.

Standards are aggregated in profiles. A standards profile is a set of standards for a particular purpose, covering certain services in the C3 taxonomy, with a guidance on implementation when and where needed. As profiles serve a particular purpose, they can be used in different environments, and therefore, they are not specific to a single overarching operational or technical concept. Profiles for Binding Metadata to a Data Object may and will be reused in other profiles.

In these profiles, interoperability standards fall into four obligation categories:

- **Mandatory** - Mandatory interoperability standards must be met to enable cross-domain information sharing
- **Conditional** - Conditional interoperability standards must be present under certain specific circumstances
- **Recommended** - Recommended interoperability standards may be excluded for valid reasons in particular circumstances, but the full implications must be understood and carefully weighed
- **Optional** - Optional interoperability standards are truly optional

The Binding Profiles use only recognized international and industry standards. The standards used are consistent with the use already declared by other services.

The Binding Profiles employ modular techniques and are extensible to provide agility in adapting to new use cases or scenarios. In other words, these profiles are designed to support the binding of any metadata to any type of finite data object.

These profiles support improved interoperability by providing a standard method to bind metadata to data objects.

1.1 Relationship to the NATO C3 Taxonomy

Due to the generic nature of the binding activity, there are multiple options for its location within the taxonomy including:

1. Metadata Binding as a CIS Security service within Core Services;
2. Metadata Binding as a Distributed service within Core Services; and
3. Metadata Binding as an organic functionality within the services originating the information.

2. Binding Concepts

The binding concepts, approaches, information, management and applications are presented and described in existing specifications for the binding mechanism (Reference [3]) and (Reference [1]). These concepts are used throughout the description of the binding profiles.

3. Conformance and Interoperability

The profiles referenced in this document are methods of applying the binding mechanism stipulated in STANAG 4778 (Reference [3]). Conformance to these profiles would determine whether an implementation adheres to the features and framework of the STANAGs and the Binding Profiles. Traditionally implementers wishing to submit an implementation to conformance testing would be responsible for:

- Preparation of a Protocol Implementation Conformance Statement (PICS) against STANAG 4778;
- Preparation of the Protocol Implementation eXtra Information for Testing (PIXIT);
- Provide input to Test Plans and Procedures;
- Approve Test Cases;
- Provide input to and approve Test Scripts; and
- Provide the Implementation Under Test (IUT).

Conformance testing of these Binding Profiles may be performed by any authorized laboratory which provides a reference implementation of the Binding Profiles. For example, the NATO C&I Agency has several reference implementations for various standards and services where the Independent Verification and Validation (IV&V) team can perform such testing¹. Although a formal Reference facility for testing of external implementations of STANAG 4778 and these Binding Profiles is not yet established, a reference implementation for STANAG 4778 has been developed and the STANAG testing capability is currently under investigation.

The outcome of formal testing ensures that the exclusive requirements of the Binding Profile under test have been properly provided and that no optional requirement impacts the expected operation nor generates an error if received by a consumer that does not implement the optional requirement.

The Interoperability Capability Team (IP Cat) will oversee the approval of test plans and procedures to be followed for the testing of these Binding Profiles.

In development of test plans, consideration will be given to assure that the implementation under test is protected, and that representatives of the originating and/or the sponsoring nation may be present while the implementation is being tested. Consideration will also be given in the test plans and procedures to protect any national or other proprietary techniques or information that may be present in an implementation submitted for compliancy or interoperability testing.

4. Interoperability Validation

The Binding Profiles have all been validated at the Coalition Warrior Interoperability eXploration, eXperimentation, eXamination, eXercise (CWIX) and/or the Technology for Information, Decision and Execution superiority (TIDE) validation exercises. These events are used for interoperability testing rather than conformance testing and serve to verify the behaviour of an implementation against an agreed reference implementation i.e. an implementation is able to

¹ The IV&V team at NCIA operates the Coalition Interoperability Assurance and Validation (CIAV) facility and the Coalition Validation and Verification Environment (CV2E) infrastructure.

interoperate with other conformant implementations. The methodology followed is standardised by the CWIX or TIDE events.

In some cases the profiles were provided as reference implementations to validate partner implementations and in other cases they were provided as an Implementation Under Test (IUT) to validate against a partner reference implementation. In each scenario agreed interoperability test cases and scripts are executed between participants.

CWIX is a yearly transformation activity to validate and improve interoperability of NATO and national C4ISR systems. CWIX is approved and supported by the Military Committee and C3 Board and operated by ACT.

The most recent validation of these Binding Profiles was performed at CWIX 2017 where the following were validated:

- Cryptographic Artefact Binding Profiles;
- Simple Mail Transfer Protocol Binding Profile;
- Simple Object Access Protocol Binding Profile;
- Representational State TransfER Binding Profile;
- Extensible Metadata Platform Binding Profile;
- Web Service Messaging Profile Binding Profile;
- [Extensible Metadata Platform Binding](#) Profile;
- Sidecar Files Binding Profile; and,
- Office Open XML Formats Binding Profile.

The following profiles were validated at CWIX 2016, CWIX 2015 or TIDE Sprint events during 2015:

- Cryptographic Artefact Binding Profiles;
- Simple Object Access Protocol Binding Profile;
- Representational State TransfER Binding Profile;
- Extensible Message and Presence Protocol Binding Profile;
- Generic Open Packaging Convention Binding Profile; and
- Sidecar Files Binding Profile.

The results of the validation efforts are provided at the [CWIX portal](#) and documented in the CWIX 2016 and 2017 Final Reports with individual results for each test case provided in [Observation, Discussion, Conclusion and Recommendation \(ODCR\) reports](#).

5. Configuration Management and Governance

Binding Profiles describe how to apply the binding mechanism specified in STANAG 4778 (Reference [3]) to specific data formats and protocols. The purpose of the Binding Profiles is to determine which of the three binding approaches (Embedded, Encapsulated, and Detached in Reference [3]) shall be best used. They specify how the BDO will be stored and transmitted for a specific data format or protocol leveraging native support, if available and they specify the semantics required to further interpret the relationship between the data object and the metadata label.

As technology evolves new data formats and protocols emerge whilst others are deprecated. Therefore, Binding Profiles may also need to evolve. It is recommended that Binding Profiles are regularly reviewed for applicability and new Binding Profiles are specified to support evolving technologies.

These Binding Profiles will be stipulated for use with both common-funded and federated systems. They will be used to promote interoperability and thus governed by the NATO and/or national authorities for interoperability.

The IP CaT provides configuration management for the NISP content and thus provide configuration management for Binding Profiles contained within this document.

6. References

- [1] AC/322(CP/1)WP(2014)0002 and AC/322(CP/4)WP(2014)0001, “Technical Standard for Confidentiality Labelling of NATO Information”, July 2014
- [2] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
- [3] STANAG 4778: Metadata Binding Mechanism, Brussels, Belgium

This page is left blank intentionally

ANNEX A: CRYPTOGRAPHIC ARTEFACT BINDING PROFILES

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

Table A-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 (appendices) 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.

2. Identification

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

Table A-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 A-2.

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

3. 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-xmlsig-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/xmlsigcore-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-xmlsig-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

This page is left blank intentionally

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.

Error! Reference source not found. Figure A-1 illustrates the structure of an XML Signature element including the primary sibling elements: SignedInfo; SignatureValue; KeyInfo; and, Object.

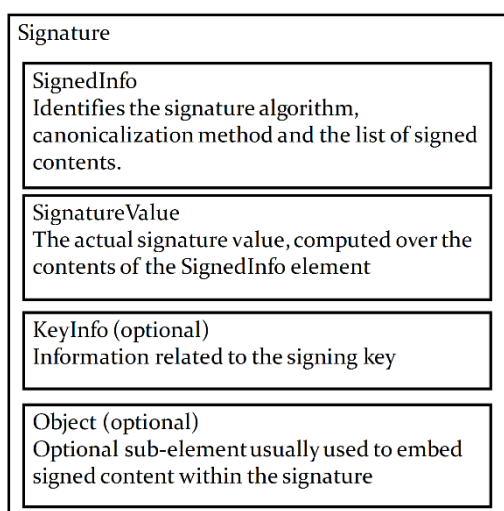


Figure A-1: XML Signature Structure

This Appendix will use the same structure as illustrated in Figure A-1 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 `SignatureMethodAlgorithm` 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 A-1-1.

Table A-1-1: 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 A-1-2.

Table A-1-2: SignatureMethod (PKI) Algorithm Identifiers

Algorithm Identifier	Mandatory/Optional/Prohibited
http://www.w3.org/2000/09/xmlsig#dsa-sha1	Prohibited
http://www.w3.org/2009/xmlsig11#dsa-sha256	Optional
http://www.w3.org/2001/04/xmlsig-more#rsa-md5	Prohibited
http://www.w3.org/2000/09/xmlsig#rsa-sha1	Prohibited
http://www.w3.org/2001/04/xmlsig-more#rsa-sha224	Optional
http://www.w3.org/2001/04/xmlsig-more#rsa-sha256	Mandatory
http://www.w3.org/2001/04/xmlsig-more#rsa-sha384	Optional
http://www.w3.org/2001/04/xmlsig-more#rsa-sha512	Optional
http://www.w3.org/2001/04/xmlsig-more#rsa-ripemd160	Optional
http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha1	Prohibited
http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha224	Optional
http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha256	Mandatory
http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha384	Optional
http://www.w3.org/2001/04/xmlsig-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 A-1-3.

Table A-1-3: SignatureMethod (HMAC) Algorithm Identifiers

Algorithm Identifier	Mandatory/Optional/Prohibited
http://www.w3.org/2000/09/xmlsig#hmac-sha1	Prohibited
http://www.w3.org/2001/04/xmlsig-more#hmac-sha224	Optional
http://www.w3.org/2001/04/xmlsig-more#hmac-sha256	Mandatory
http://www.w3.org/2001/04/xmlsig-more#hmac-sha384	Optional
http://www.w3.org/2001/04/xmlsig-more#hmac-sha512	Optional
http://www.w3.org/2001/04/xmlsig-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>9JBAVs2gUWUzFh8uU1lubXW13VgQxli3NM+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+8lJt4nv4GyRNC= </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 mb: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 A-2: 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>fAXcjRa4z1LyB+lchyBK/9Jz1soZSbxNCmr/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>j5AgAamc6cv54VDz10kDlQ4wYZLLAU3761eFOUWvtX0=</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>gItAuwdEykw5x5Dht50TOeilxfT0q7KLaUXm4w/2rnpTjoxiODTI3Wr8D4fmx/404bVrX23S
tY6HHT/dxDPcgODa+K9YL/pl3y8RvIrfWghiZReY5AUj1EF3mxI22ari/ao0shKe18aPJ0J2RmGH3t30qrHfvUX
cIcREIOT1S6GajpNCOJPYoa9yb400MOx0oRHXXfegNq5eXesBIh2u4DhwL0I4GSeuYA9Fvt8qyv1a9EnTTS6fG2
+gLjd6YEQzfIBvVtrY5b9WnhqqiHy5tyepZgVtMSEXrukWrNELpvc467KR+MincgUA9RlsAEvCBAR4oQKTUOxB
Q5tD+N/FzQ==</SignatureValue>
    <KeyInfo Id="id-9920a48c-c3a1-45d0-a81c-1ce04d1d8de6">
      <X509Data>
        <X509Certificate>MIIDM.....wIBAgIAI29/+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 mb: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 A-3: 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"
/>
      </SignedInfo>
      <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>UbMTebL9lKFARnGlqWOpQ1DiuCFPzs6Wlhse9gPOxUk=</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/wKZjOIDVGGyvlrigTDLj6gRsQCTHY=</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 mb: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 mb:URI="" />
  </mb:MetadataBinding>
</mb:MetadataBindingContainer>
</BindingInformation>
</Document>

```

Figure A-4: 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>0JsT5SNKuCYoe91t18n590Hcy/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>a3yUgG8j0eIPi6ZSw7aw4JPH01SBglS0+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+OtgPY8+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>ClyPwzpU/ngO42sXo2HHZTtbXNte2FAXf2RivMy5u6z/xoNlmi/mHm5ejZPFWkoGaUmWDad
REcc51i6XBYXeks2YVyMh05uDRCLPYnKIAX3BpUFH7y9JUKlj4WvldBeZ2GwNhp463QMvn8pf35cXwlf86VcOM
3CtAm5MNbnS6BqgsdwygCF/HivjHcQSnYGRhI4vegelwFYyhFRHQ1OE3ytUDR8VLKZfgYK3M6mcQjv1HtL2qjR
xMHRkQQt8oBQk6iAWxYgbqeIzqw3cIYL5jb/ML2UOycGgwUIqGFx95EouKuOMZSN8e2dnaVaHp26X1zpdJkyTk
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 mb: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 mb:URI="#para-2-1" />
</mb:MetadataBinding>
</mb:MetadataBindingContainer>
</mb:BindingInformation>
</Document>

```

Figure A-5: Embedded Cryptographic BDO Containing a Detached Signature with a Digital Signature Cryptographic Artefact

ANNEX B: SIMPLE MAIL TRANSFER PROTOCOL BINDING PROFILE

1. SMTP Introduction

This profile specifies the mechanism for binding metadata to MIME entities, such as internet mail messages (and sub-parts thereof). A MIME entity can be a sub-part, sub-parts of a message or the message with all its sub-parts. A MIME entity that is the message includes only the MIME message headers and MIME body (Reference [6]), and does not include the internet email headers (Reference [3]).

This profile does not support the capability for referencing internet email headers (or subsets thereof). A separate profile will specify how to bind metadata to internet email headers.

2. Identification

The profile for SMTP is uniquely identified by the Canonical Identifier shown in Table B-1.

Table B-1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:smtp
Version Identifier	urn:nato:stanag:4778:profile:smtp:1:1

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

- changes to the base SMTP standards
- 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 B-1.

This document deprecates the previous version identified by Version Identifier urn:nato:stanag:4778:profile:smtp:1:0.

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

3. Standards

- [1] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
- [2] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
- [3] IETF RFC 5322, "Internet Message Format", at <http://tools.ietf.org/html/rfc5322>, October 2008.
- [4] IETF RFC 7444, "Security Labels in Internet Email", K. Zeilenga and A. Melnikov, at <http://tools.ietf.org/html/rfc7444>, February 2015.
- [5] IETF RFC 2392, "Content-ID and Message-ID Uniform Resource Locators", at <http://tools.ietf.org/html/rfc2392>, August 1998.
- [6] IETF RFC 2045, "Multipurpose Internet Mail Extensions(MIME) Part One: Format of Internet Message Bodies", at <http://tools.ietf.org/html/rfc2045>, November 1996

- [7] IETF RFC 2231, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", at <http://tools.ietf.org/html/rfc2231>, November 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

4. Notational Conventions

- 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 SIO²-Label (Reference [4]), Message-ID ((Reference [5])) and Content-ID (Reference [5]) URI schemes and MIME Entities (Reference [6]).

5. Internet Email Structure

The BDO is a detached BDO that MUST contain at least one *MetadataBinding* that contains a *DataReference URI* attribute value conformant with the Message-ID Uniform Resource Locator, `mid`, scheme according to (Reference [5]). By conforming to Reference [5] to syntactically and semantically interpret the *DataReference URI* attribute allows for the metadata to be bound to the entire message. For the purposes of this profile the entire message is a MIME Entity that consists of the MIME message headers and the MIME body (Reference [6]).

The *DataReference xmime:contentType* attribute value is REQUIRED when the *URI* attribute value is the `mid` URI scheme.

The *DataReference xmime:contentType* attribute value SHALL be *message/rfc822* when the *URI* attribute value is the `mid` URI scheme.

This profile requires that the SIO-Label header field as specified in (Reference [4]) SHALL be used to embed the BDO within the internet mail message.

The BDO MUST be included in the SIO-Label header `label` parameter.

The SIO-Label `label` parameter value MUST be the `base64` encoding of the BDO.

The SIO-Label `type` parameter MUST be present with the value *urn:nato:stanag:4778:bindinginformation:1:0*.

It must be noted that the `label` parameter SHALL conform to Reference [7] (as specified in Reference [4]) specifically in relation to parameter value continuation.

Depending upon the line length limit (recommended to be 78 characters or less and not more than 998 characters – see Reference [3]) the `label` parameter SHALL be split into multiple `label` parameters, as illustrated below.

² SIO stands for Security Information Object, as defined in X.841

```
label*0="PFNIY0xhYmVsIHhtbG5zPSJodHRwOi8vZXhhbX";  
label*1="BsZS5jb20vc2VjLWxhYmVsLzAiPjxQb2xpY3Ij";  
label*2="ZGVudGlmaWVyIFVSST0idXJuOm9pZDoxLjEiLz";  
label*3="48Q2xhc3NpZmljYXRpb24+MzwvQ2xhc3NpZmlj";  
label*4="YXRpb24+PC9TZWNMYWJlD4=";
```

It is noted that SIO-Label `label` parameter value production implicitly allows for white space (e.g. folding) as described in Reference [4] Section 4. As such, implementations SHALL be able to process SIO-Label `label` parameter values that contain white space as illustrated below:

```
label*0="PFNIY0xhYmVsIHhtbG5zPSJodHRwOi8vZXhhbXBsZS5jb20vc2VjLW  
xhYmVsLzAiPjxQb2xpY3Ij";  
label*1="ZGVudGlmaWVyIFVSST0idXJuOm9pZDoxLjEiLz48Q2xhc3NpZmlj  
YXRpb24+MzwvQ2xhc3NpZmlj";  
label*2="YXRpb24+PC9TZWNMYWJlD4=";
```

It is also noted that Reference [7] allows for quoted-string values (for parameter production). As such, implementations SHALL be able to process SIO-Label `label` parameter values that contain quoted-string values.

An example of an Embedded BDO contained in the SIO-Label header field of an internet mail message that illustrates the binding of Confidentiality Metadata Labels (Reference [1]) as example metadata to the message is provided in Figure B-1.

```

From: alan.ross@smhs.co.uk
To: alan.ross@reach.nato.int
SIO-Label: type="urn:nato:stanag:4778:bindinginformation:1:0"; label=<base64 BIO>
Message-Id: <unique-msg-id@smhs.co.uk>

This is a simple informal message


<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"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
  <mb:MetadataBindingContainer>
    <mb:MetadataBinding>
      <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="mid://unique-msg-id@smhs.co.uk"
        xmime:contentType="message/rfc822"/>
      </mb:MetadataBinding>
    </mb:MetadataBindingContainer>
  </mb:BindingInformation>

```



Figure B-1: Example of Binding Confidentiality Metadata Label to Email

In the case where metadata is to be bound to individual MIME bodyparts, the *URI* attribute of the *DataReference* element **MUST** use the Content-ID Uniform Resource Locator, *cid*, scheme according to (Reference [5]).

The MIME Content-Type header field value, that indicates the internet media type of the MIME bodypart, **SHALL** be used as the *DataReference xmime:contentType* attribute value.

The example provided in Figure B-2 illustrates an Embedded BDO contained in the SIO-Label header field of an internet mail message where Confidentiality Metadata Labels (Reference [1]) as example metadata are bound to:

1. a message; and
2. a MIME bodypart included in the message.


```

From: alan.ross@smhs.co.uk
To: alan.ross@reach.nato.int
SIO-Label: type="urn:nato:stanag:4778:bindinginformation:1:0"; label=<base64 BIO>
Message-Id: <unique-msg-id@smhs.co.uk>
Content-Type: multipart/mixed;
    boundary="boundary-001";

--boundary-001

Content-ID: <unique-content-id-001@smhs.co.uk>
Content-Type: application/pdf;

..etc..

--boundary-001--
  <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"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
    <mb:MetadataBindingContainer>
      <mb:MetadataBinding>
        <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="mid://unique-msg-id@smhs.co.uk"
          xmime:contentType="message/rfc822"/>
        </mb:MetadataBinding>
      <mb:Metadata>
        <slab:originatorConfidentialityLabel
          xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
          <slab:ConfidentialityInformation>
            <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
            <slab:Classification>RESTRICTED</slab:Classification>
          </slab:ConfidentialityInformation>
          <slab:CreationDateTime>
            2015-09-30T12:30:00Z
          </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
      </mb:Metadata>
      <mb:DataReference
        URI="cid://unique-content-id-001@smhs.co.uk"
        xmime:contentType="application/pdf"/>
      </mb:MetadataBinding>
    </mb:MetadataBindingContainer>
  </mb:BindingInformation>

```



Figure B-2: Example Binding of Confidentiality Metadata Labels to Email Message and Attachment

6. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Annex A) SHALL be adhered to for the use cases that cryptographic bindings are required to provide a higher level of integrity protection, authenticity and non-repudiation of the binding specified in this profile.

Unless otherwise stated, all statements that apply to the Cryptographic Artefacts binding profile (Annex A) also apply to this profile for generating and validating cryptographic bindings.

The creation of the `DigestValue` specific to this profile SHALL conform to the following rules for XML Signature Core Generation:

1. For each MIME entity that is referenced within a metadata binding (either the message or individual MIME bodyparts) that MIME entity SHALL be canonicalised according to Reference [8] Section 3.1.1.
2. The canonicalised MIME entity SHALL be input to the `DigestMethod` Algorithm.

The creation of the `DigestValue` specific to this profile SHALL conform to the following rules for XML Signature Core Validation:

1. For each `Reference` in the `Manifest` that dereferences a MIME entity (either the message or individual MIME bodyparts) that MIME entity SHALL be canonicalised according to Reference [8] Section 3.1.1.
2. The canonicalised MIME entity SHALL be input to the `DigestMethod` Algorithm.

ANNEX C: EXTENSIBLE MESSAGE AND PRESENCE PROTOCOL BINDING PROFILE

1. XMPP Introduction

Confidentiality metadata labels can be supported in XMPP stanzas as indicated by XEP-0258 (Reference [4]) whereby a mechanism for carrying Enhanced Security Services (ESS) Security labels (Reference [1]) is standardized. This profile extends the XEP-0258 (Reference [4]) specification to support carrying an Embedded or Detached BDO for `Message` stanzas. This profile supports the XMPP use cases for one-to-one instant messaging and multi-user chat.

Future profiles for XMPP will specify support for carrying BDOs in `IQ` stanzas specifically to support Publish Subscribe mechanisms such as those defined in XEP-0060 Publish-Subscribe (Reference [5]).

2. Identification

The profile for XMPP is uniquely identified by the Canonical Identifier shown in Table C-1.

Table C-1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:xmpp
Version Identifier	urn:nato:stanag:4778:profile:xmpp:1:1

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

- changes to the base XMPP standard
- 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 C-1.

This document deprecates the previous version identified by Version Identifier urn:nato:stanag:4778:profile:xmpp:1:0.

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

3. Standards

- [1] IETF RFC 2634, “Enhanced Security Services for S/MIME”, at <http://tools.ietf.org/html/rfc2634>, June 1999.
- [2] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
- [3] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
- [4] XEP-0258, “Security Labels in XMPP”, version 1.1, at <http://www.xmpp.org/extensions/xep-0258.html>, April 2013
- [5] XEP-0060, “Publish-Subscribe”, version 1.3, at <http://www.xmpp.org/extensions/xep-0060.html>, July 2010

- [6] IETF RFC 6122, “Extensible Messaging and Presence Protocol (XMPP): Address Format”, at <http://tools.ietf.org/html/rfc6122>, March 2011
- [7] IETF RFC 6120, “Extensible Messaging and Presence Protocol (XMPP): Core”, at <http://tools.ietf.org/html/rfc6120>, March 2011
- 4. IETF RFC 6121, “Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence”, at <http://tools.ietf.org/html/rfc6121>, March 2011

Notational Conventions

- 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 [3].
- Courier font indicates syntax derived from XMPP (References [7], 4 and [6]) and XEP-0258 (Reference [4]).

5. Message Stanza Structure

The Message stanza structure is specified in (Reference 4). Dependent upon system information exchange requirements it may be necessary that the Message stanza is bound to the metadata or subsets of the Message stanza are bound to the metadata. As such, Binding Information SHALL be represented either as: an Embedded BDO; or, a Detached BDO.

Figure C-1 illustrates the high-level structure of a Message stanza that contains an Embedded BDO contained within a XEP-0258 `securitylabel` element.

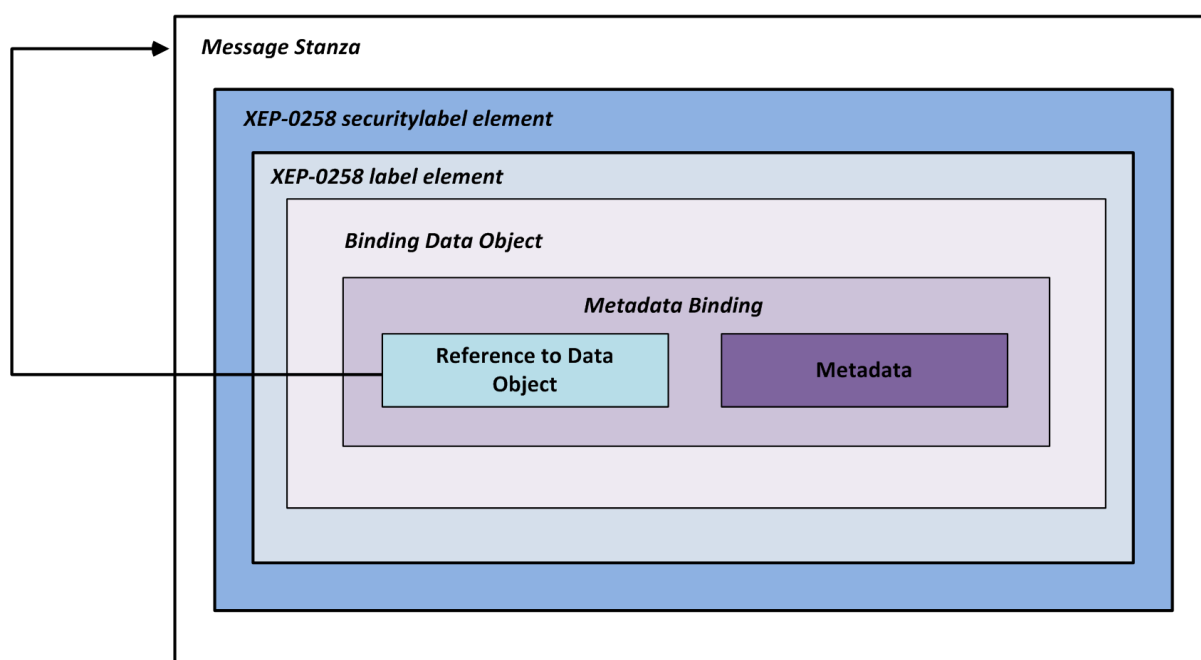


Figure C-1: Structure of Message Stanza Containing Embedded BDO

The BDO SHALL be contained in a `label` child element of a XEP-0258 `securitylabel` element that MUST include the *BindingInformation* element only (as a child element of the `label` element).

It is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

An example of a BDO embedded in a Message stanza that illustrates the binding of the entire Message stanza to metadata is provided in Figure C-2. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.



```
<message to="alan.ross@smhs.co.uk" from="alan.ross@reach.nato.int">
  <body>This is a labelled XMPP message</body>
  <securitylabel xmlns='urn:xmpp:sec-label:0'>
    <label>
      <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"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
        xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
        <mb:MetadataBindingContainer>
          <mb:MetadataBinding>
            <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:MetadataBindingContainer>
      </mb:BindingInformation>
    </label>
  </securitylabel>
</message>
```

Figure C-2: Example Embedded Binding Data Object for Message Stanza (XMPP)

An example of a detached BDO contained in a Message stanza that illustrates the binding of the body element (child of the Message stanza) to metadata is provided in Figure C-3. This example illustrates the use of XPath for referencing the body element. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

Note in Figure C-3 the *namespace-uri* attribute value is set as `jabber:client`. In XMPP, stanzas may belong to different XMPP content namespaces i.e. `jabber:client` and `jabber:server` depending on whether the XMPP stream is negotiated between an XMPP client and XMPP server or an XMPP server and a peer XMPP server, respectively. The only difference between the two is that the `to` and `from` attributes are optional on stanzas qualified by the `jabber:client` namespace and required on stanzas qualified by the `jabber:server` namespace. To accommodate the re-scoping of XMPP content namespaces as described above the following rules apply:

1. If the XMPP Binding Profile is supported only between XMPP peer servers the *namespace-uri* attribute value SHALL be `jabber:server`; otherwise,
2. The default *namespace-uri* attribute value SHALL be `jabber-client`.

```

<message to="alan.ross@smhs.co.uk" from="alan.ross@reach.nato.int">
  <body>This is a labelled XMPP message</body>
  <securitylabel xmlns='urn:xmpp:sec-label:0`'>
    <label>
      <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"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
        xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
        <mb:MetadataBindingContainer>
          <mb:MetadataBinding>
            <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="">
              <ds:Transforms>
                <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
                  <ds:XPath>
                    ancestor-or-self::*[local-name()='body' and namespace-uri()='jabber:client']
                  </ds:XPath>
                </ds:Transform>
              </ds:Transforms>
            </mb:DataReference>
          </mb:MetadataBinding>
        </mb:MetadataBindingContainer>
      </mb:BindingInformation>
    </label>
  </securitylabel>
</message>

```



Figure C-3: Example Detached Binding Data Object Contained in Message Stanza (XMPP)

6. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Annex A) SHALL be adhered to for the use cases that cryptographic bindings are required to provide a higher level of integrity protection, authenticity and non-repudiation of the binding specified in this profile.

Unless otherwise stated, all statements that apply to the Cryptographic Artefacts binding profile (Annex A) also apply to this profile for generating and validating cryptographic bindings.

ANNEX D: OFFICE OPEN XML FORMATS BINDING PROFILE

1. OOXML Introduction

The Office Open XML Formats (OOXML) are defined ISO/IEC 29500 (Reference [1]) and offer standards for representing office documents, including spreadsheets, presentations and word processing documents.

OOXML adopts a structured format which consists of a number of XML-based files packaged into an archive file according to the Open Packaging Conventions (OPC), which is defined in Part 2 of ISO/IEC 29500 (Reference [1]).

OOXML allows for custom XML files to be included within the package without impacting the underlying application. This provides a mechanism for a metadata to be bound to the OOXML document and maintained within the package.

This profile for the OOXML describes how metadata can be maintained.

2. Identification

The profile for OOXML is uniquely identified by the Canonical Identifier shown in Table D-1.

Table D-1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:ooxml
Version Identifier	urn:nato:stanag:4778:profile:ooxml:1:1

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

- changes to the base OOXML standard e.g.
 - introduction of new package parts
- additional profiles for OOXML e.g.
 - different combinations of package parts
 - bindings to elements within a package part (e.g. binding metadata to paragraphs within a document)
- 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 D-1.

This document deprecates the previous version identified by Version Identifier urn:nato:stanag:4778:profile:ooxml:1:0.

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

3. Standards

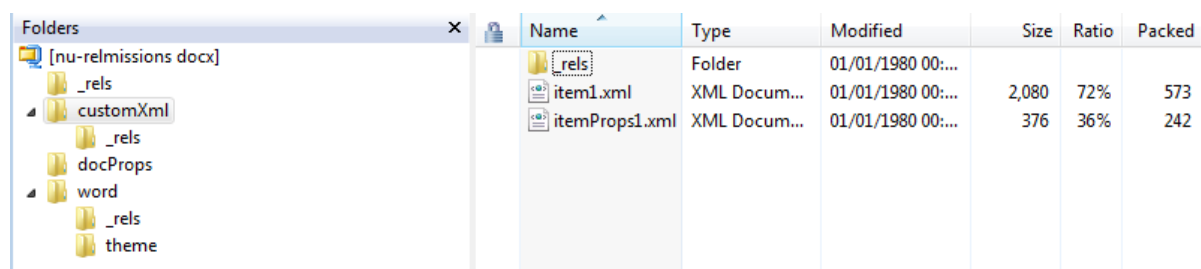
- [1] ISO/IEC 29500-2 “Office Open XML File Formats - Part 2: Open Packaging Conventions”, at http://standards.iso.org/ittf/PubliclyAvailableStandards/c061796_ISO_IEC_29500-2_2012.zip, August 2012
- [2] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
- [3] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium

4. Notational Conventions

- 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 [3].

5. Structure

The structure of an OOXML package consists of a number of folders which contain different components of the document.



Name	Type	Modified	Size	Ratio	Packed
_rels	Folder	01/01/1980 00:...			
item1.xml	XML Docum...	01/01/1980 00:...	2,080	72%	573
itemProps1.xml	XML Docum...	01/01/1980 00:...	376	36%	242

Figure D-1: General Structure of an OPC Package

The structure, as shown in Figure D-1, generally consists of:

- An application specific folder, for example “word”, “ppt” or “xl”.
- A “customXml” folder in which arbitrary XML files can be stored.
- A “docProps” folder in which core and custom document properties are held.
- Multiple “_rels” folder which contains details of the parts within a folder.

This structure is then packaged into an archive file with an application specific extension (for example, .docx).

The document that is displayed to a user is generally split over a number of different XML files contained with the package. This does not present a problem when applying granular metadata to different parts of the document.

However care must be taken when the intention is to bind metadata to the complete document (refer to Microsoft Office File Types section below for normative text related to binding metadata to a whole document). For example, the XML file /word/document.xml within a Microsoft Word OPC package does not contain the headers or footers of the document (these are contained in the separate files /word/header1.xml and /word/footer1.xml.)

6. Custom XML

In order to support metadata binding within an OPC package, a single CustomXML file SHALL be maintained within the OPC package with the Metadata Binding Container namespace, “urn:nato:stanag:4778:bindinginformation:1:0”.

DataReference elements SHALL be used to reference the files within the OPC package.

Data elements SHALL NOT be used.

DataReference elements used to reference the files within the OPC package will use the Pack URI scheme ‘pack’ as specified in Reference [1] Annex B.

The authority component of the Pack URI scheme SHALL be empty that denotes the package root.

When referring to files, or portions of files, within the OPC package, absolute URIs from the package root SHALL be used with the *DataReference* element. For example,

```
<DataReference URI="pack:///word/document.xml"/>
```

Microsoft Office File Types

Microsoft Office has used the OOXML standard, since Microsoft Office 2007, for a number of its document types, including Microsoft Word, Microsoft Excel and Microsoft PowerPoint.

When binding metadata to a complete document (as opposed to a specific part of a document), all of the files (when they are present within the package) listed in the “Package File” for the particular document type SHALL be referenced in the binding (see in Table D-2).

Table D-2: Packages Files to be Referenced in a Binding to a Complete Document³

Application	Package File	Description
Microsoft Word	/word/document.xml	The document.
	/word/styles.xml	The styles within the document.
	/word/header<N>.xml	The headers for sections within the document.
	/word/footer<N>.xml	The footers for sections within the document.
	/word/media/*	The media (e.g. pictures) embedded in the document.
	/word/footnotes.xml	The footnotes.
	/word/endnotes.xml	The endnotes.
	/word/comments.xml	The review comments.
	/word/commentsExtended.xml	The review comments.
Microsoft Excel	/xl/workbook.xml	The workbook.
	/xl/styles.xml	The styles within the workbook.
	/xl/sharedStrings.xml	The strings shared between worksheets.
	/xl/worksheets/sheet<N>.xml	The worksheets within the workbook.
	/xl/charts/chart<N>.xml	The charts on a worksheet.
	/xl/charts/colors<N>.xml	The colors of a chart on a worksheet.
	/xl/charts/styles<N>.xml	The style of a chart on a worksheet.
	/xl/pivotTables/pivotTable<N>.xml	The pivotTables on a worksheet.
	/xl/comments<N>.xml	The comments on a worksheet.
Microsoft PowerPoint	/ppt/presentation.xml	The presentation.
	/ppt/slides/slide<N>.xml	The slides within the presentation.
	/ppt/slideLayouts/slideLayout<N>.xml	The slide layouts.
	/ppt/slideMaster/slideMaster<N>.xml	The slide masters.
	/ppt/comments/comment<N>.xml	The comments on a slide.
	/ppt/media/*	The media (e.g. pictures) embedded on the slides.
	/ppt/presProps.xml	The additional presentation-wide properties.
	/ppt/viewProps.xml	The additional presentation-wide properties.

The common document properties package files (where they are present within the package) SHALL also be referenced in the binding (see Table D-2).

Additional package files, beyond those listed in Table D-2 and Table D-3, MAY be referenced in the binding (e.g. packages files created by COI-specific Office Add-Ins).

Table D-3: Common Packages Files to be Referenced in a Binding to a Complete Document

Package File	Description
/docProps/core.xml	The common document properties.
/docProps/app.xml	The application-specific document properties.
/docProps/custom.xml	The custom (e.g. user defined) document properties.

³ The notation "<N>" in the "Package File" column indicate an increasing integer. For example, "/word/header<N>.xml" would indicate the package files "/word/header1.xml" and "/word/header2.xml" in a document with two headers.

Figure D-2 shows the contents of a CustomXML file, stored in /customXml/item1.xml, for a simple Microsoft Word document containing an embedded image. It uses Confidentiality Metadata Labels (Reference [2]) as example metadata.

```
<mb:BindingInformation
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
  <mb:MetadataBindingContainer>
    <mb:MetadataBinding>
      <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>2016-11-10T12:30:00Z</slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
      </mb:Metadata>
      <mb:DataReference URI="pack:///word/document.xml"/>
      <mb:DataReference URI="pack:///word/styles.xml"/>
      <mb:DataReference URI="pack:///word/header1.xml"/>
      <mb:DataReference URI="pack:///word/footer1.xml"/>
      <mb:DataReference URI="pack:///word/media/image.jpeg"
xmime:contentType="image/jpeg"/>
      <mb:DataReference URI="pack:///word/footnotes.xml"/>
      <mb:DataReference URI="pack:///word/endnotes.xml"/>
      <mb:DataReference URI="pack:///docProps/app.xml"/>
      <mb:DataReference URI="pack:///docProps/core.xml"/>
      <mb:DataReference URI="pack:///docProps/custom.xml"/>
    </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
</mb:BindingInformation>
```

Figure D-2: CustomXML file

7. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Annex A) SHALL be adhered to for the use cases that cryptographic bindings are required to provide a higher level of integrity protection, authenticity and non-repudiation of the binding specified in this profile.

Unless otherwise stated, all statements that apply to the Cryptographic Artefacts binding profile (Annex A) also apply to this profile for generating and validating cryptographic bindings.

This page is left blank intentionally

ANNEX E: SIMPLE OBJECT ACCESS PROTOCOL BINDING PROFILE

1. SOAP Introduction

It is recognised that service providers and service consumers implementing web services based on SOAP operate under different frameworks and application contexts. As such, this profile includes support for both SOAP 1.1 (Reference [3]) and SOAP 1.2 (Reference [4]). To support information sharing between partners it may be necessary to locate a Binding Data Object (BDO) in the SOAP protocol layer. Metadata may be bound to the whole data object (SOAP message) or may be bound to subsets of the SOAP message (data object(s) in the SOAP body). Where there is a requirement to bind metadata to a SOAP message or data object (s) within the SOAP body that is exchanged between a service consumer and a service provider, the SOAP Binding Profile specified must be adhered to.

2. Identification

The profile for SOAP is uniquely identified by the Canonical Identifier shown in Table E-1.

Table E-1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:soap
Version Identifier	urn:nato:stanag:4778:profile:soap:1:1

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

- changes to the base SOAP standard
- 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 E-1.

This document deprecates the previous version identified by Version Identifier urn:nato:stanag:4778:profile:soap:1:0.

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

3. Standards

- [1] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
- [2] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
- [3] W3C SOAP Version 1.1, 2000, "Simple Object Access Protocol (SOAP 1.1)", at <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>, W3C Recommendation, W3C, 8 May 2000.

- [4] W3C SOAP Version 1.2, 2007, "SOAP Version 1.2", at <http://www.w3.org/TR/soap12-part1/>, W3C Recommendation, W3C, 27 April 2007.
- [5] W3C XMLDSIG-CORE, 2008, "XML- Signature Syntax and Processing (Second Edition)", at <http://www.w3.org/TR/2008/REC-xmlsig-core-20080610/>, W3C Recommendation, W3C, 10 June 2008

4. Namespace Constraints

The table below summarises the XML namespaces and corresponding prefixes used throughout for the binding of metadata to SOAP data objects and portions thereof.

Table E-2: XML Namespaces and Prefixes

Prefix	Namespace
mb	urn:nato:stanag:4778:bindinginformation:1:0
soap	http://schemas.xmlsoap.org/soap/envelope/ or http://www.w3.org/2003/05/soap-envelope
soap11	http://schemas.xmlsoap.org/soap/envelope/
soap12	http://www.w3.org/2003/05/soap-envelope
wsa	http://www.w3.org/2005/08/addressing
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
wsse11	http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd
xs	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance

5. Notational Conventions

- 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 [3].
- Courier font indicates syntax derived from various W3C XML Signature (Reference [5]) and SOAP (References [3], [4]) standards.

6. SOAP Message Structure

The SOAP message structure is specified in (References [3], [4]). Dependent upon system information exchange requirements it may be necessary that the whole SOAP message is bound to the metadata or subsets of the SOAP message are bound to the metadata. As such, Binding Information SHALL be represented either as: an Embedded BDO; or, a Detached BDO.

The BDO is contained in a Security header that SHALL include the *BindingInformation* element only (as a child element of the Security element).

If the SOAP message is SOAP 1.1 the Security @actor attribute SHALL be included with a value of *urn:nato:stanag:4778:bindinginformation:1:0:role:bindingInformationReceiver*.

If the SOAP message is SOAP 1.2 the Security @role attribute SHALL be included with a value of *urn:nato:stanag:4778:bindinginformation:1:0:role:bindingInformationReceiver*.

It is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

An example of a BDO embedded in a SOAP 1.1 message that illustrates the binding of the SOAP message to metadata is provided in Figure E-1. Also illustrated is the use of the `actor` attribute to support multiple Security elements. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```
<soap11:Envelope xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/">
  <soap11:Header>
    <wsse:Security
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd"
      soap11:actor="
urn:nato:stanag:4778:bindinginformation:1:0:role:bindingInformationReceiver">
      <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"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <mb:MetadataBindingContainer>
          <mb:MetadataBinding>
            <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:MetadataBindingContainer>
      </mb:BindingInformation>
    </wsse:Security>
  </soap11:Header>
  <soap11:Body>
    <Track xmlns="http://example.com/trackInformation">
      ....
    </Track>
  </soap11:Body>
</soap11:Envelope>
```

Figure E-1: Example Embedded BDO for SOAP

An example of a detached BDO contained in a SOAP 1.1 message that illustrates the binding of an external data object in the SOAP body to metadata is provided in Figure E-2. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

Figure E-2 illustrates the use of XPointer and XPath to reference the data object. Also illustrated is the use of the `actor` attribute to support multiple Security elements.

```
<soap11:Envelope xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/">
  <soap11:Header>
    <wsse:Security
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd"
      soap11:actor="
urn:nato:stanag:4778:bindinginformation:1:0:role:bindingInformationReceiver ">
      <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"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <mb:MetadataBindingContainer>
          <mb:MetadataBinding>
            <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="">
    <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
        <ds:XPath>
          ancestor-or-self::*[local-name()='Track' and namespace-
uri()='http://example.com/trackInformation']
        </ds:XPath>
      </ds:Transform>
    </ds:Transforms>
  </mb:DataReference>
</mb:MetadataBinding>
</mb:MetadataBindingContainer>
</mb:BindingInformation>
</wsse:Security>
</soap11:Header>
<soap11:Body>
  <Track xmlns="http://example.com/trackInformation">
    ....
  </Track>
</soap11:Body>
</soap11:Envelope>

```

Figure E-2: Example Detached BDO for SOAP

7. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Annex A) SHALL be adhered to for the use cases that cryptographic bindings are required to provide a higher level of integrity protection, authenticity and non-repudiation of the binding specified in this profile.

Unless otherwise stated, all statements that apply to the Cryptographic Artefacts binding profile (Annex A) also apply to this profile for generating and validating cryptographic bindings.

ANNEX F: REPRESENTATIONAL STATE TRANSFER BINDING PROFILE

1. RESTful Introduction

REST is an architectural style defined as a set of constraints on a distributed hypermedia system and implemented by a set of standard protocols that adhere to these constraints. The REST architectural style can be employed for implementing web services which are known as RESTful web services. RESTful web services rely upon the Hypertext Transport Protocol (HTTP) (Reference [4]) as the standard interface between service providers and service consumers utilizing the HTTP verbs GET, PUT, POST, DELETE, etc. in their specified manner. Resources that are exposed through RESTful web services are identified by URIs and are represented to service consumers in any (mutually agreed) media type format. In other words, a URI identifies a resource, rather than a representation, and when a service consumer asks a service provider for a resource, the service provider will respond with the best possible representation for that resource, given the service consumer's preferences. In an environment where data objects must have bound metadata, the resource identified in the URI will already contain a BDO (detached, encapsulating or embedded). As such, there is no requirement for metadata binding that is specific for REST. However, to support information sharing between partners it may be necessary to locate a Binding Data Object (BDO) in the HTTP protocol layer.

This profile specifies the mechanism for binding metadata to the HTTP Entity message body (Reference [4] Section 3.3).

This profile does not support the capability for referencing HTTP Entity message start line (Reference [4] Section 3.1) or HTTP Entity message headers (Reference [4] Section 3.2). A separate profile will specify how to bind metadata to HTTP Entity message start line and headers.

2. Identification

The profile for REST is uniquely identified by the Canonical Identifier shown in Table F-1.

Table F-1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:rest
Version Identifier	urn:nato:stanag:4778:profile:rest:1:1

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

- changes to the base RESTful standard
- 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 F-1.

This document deprecates the previous version identified by Version Identifier urn:nato:stanag:4778:profile:http:1:0.

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

3. Standards

- [1] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
- [2] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
- [3] IETF RFC 7444, "Security Labels in Internet Email", K. Zeilenga and A. Melnikov, at <http://tools.ietf.org/html/rfc7444>, February 2015.
- [4] IETF RFC 7230, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", at <http://tools.ietf.org/html/rfc7230>, June 2014.
- [5] IETF RFC 2231, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", at <http://tools.ietf.org/html/rfc2231>, November 1997.
- [6] ITU-T X.841, "Information Technology – Security Techniques – Security information objects for access control", at <https://www.itu.int/rec/T-REC-X.841/en>, October 2000
- [7] IETF RFC 5751, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", at <http://tools.ietf.org/html/rfc5751>, January 2010

4. Notational Conventions

- 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 [4]
- Courier font indicates syntax derived from SIO⁴-Label (Reference [3]) and HTTP (Reference [4]) referenced in this Annex.

5. HTTP Request/Response for RESTful Web Services

In the cases where there is a requirement for BDOs to be located in the HTTP protocol layer it is RECOMMENDED to use the SIO-Label (Reference [4]) as a HTTP Entity message header for HTTP Entity requests and responses for storing the BDO.

The BDO is an embedded BDO that MUST contain at least one *MetadataBinding* that contains a null *DataReference URI* attribute value (refer to Same-Document References Section of Reference [2]) that semantically indicates a binding relationship to the HTTP Entity message body request or response.

The *DataReference xmime:contentType* attribute MUST be present with a value of *message/http*.

The BDO MUST be included in the SIO-Label header `label` parameter.

⁴ SIO stands for Security Information Object, as defined in X.841 (Reference [6])

The SIO-Label label parameter value MUST be the base64 encoding of the BDO.

HTTP (Reference [4]) does not specify a line length limit for HTTP header field values and does not support parameter value continuation as specified in Reference [7]. Therefore, the SIO-Label label parameter MUST not support Reference [7] for parameter value continuation.

The SIO-Label type parameter MUST be present with the value *urn:nato:stanag:4778:bindinginformation:1:0*.

Figure F-1 illustrates an HTTP POST request with the SIO-Label HTTP header field with the header field value as specified in this Binding Profile. Figure F-2 illustrates the base64 decoded value of the label value parameter. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```
POST /token HTTP/1.1
Host: server.example.com
SIO-Label: type="urn:nato:stanag:4778:bindinginformation:1:0" label="<base64 encoded BDO>"
Content-Type: text/xml

<Document>
...
</Document>
```

Figure F-1: An example HTTP POST Request which includes an embedded BDO

```
<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"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <mb:MetadataBindingContainer>
    <mb:MetadataBinding>
      <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="" xmime:contentType="message/http"/>
    </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
</mb:BindingInformation>
```

Figure F-2: Base64 Decoded Embedded BDO illustrating the binding of the HTTP POST REQUEST

6. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Annex A) SHALL be adhered to for the use cases that cryptographic bindings are required to provide a higher level of integrity protection, authenticity and non-repudiation of the binding specified in this profile.

Unless otherwise stated, all statements that apply to the Cryptographic Artefacts binding profile (Annex A) also apply to this profile for generating and validating cryptographic bindings.

The creation of the `DigestValue` specific to this profile SHALL conform to the following rules for XML Signature Core Generation:

The HTTP Entity message body SHALL be canonicalised according to Reference [8] Section 3.1.1.

The canonicalised HTTP Entity message body SHALL be input to the `DigestMethod` Algorithm.

The creation of the `DigestValue` specific to this profile SHALL conform to the following rules for XML Signature Core Validation:

For each Reference in the Manifest that dereferences the HTTP Entity message body SHALL be canonicalised according to Reference [8] Section 3.1.1.

The canonicalised HTTP Entity message body SHALL be input to the `DigestMethod` Algorithm.

ANNEX G: GENERIC OPEN PACKAGING CONVENTION BINDING PROFILE

1. OPC Introduction

This profile defines a generic packaging mechanism, based upon the Open Packaging Container (OPC) defined in ISO/IEC 29500-2:2008 (Reference [1]), to associate any arbitrary file that do not use the Office Open XML (OOXML) format (Reference [1]) or have no specific profile for supporting the *BindingInformation* with their own file format.

In OPC terminology, the term *package* corresponds to a ZIP archive and the term *part* corresponds to a file stored within the ZIP. Every part in a package has a unique URI-compliant part name along with a specified content-type expressed in the form of a MIME media type. A part's content-type explicitly defines the type of data stored in the part, and reduces duplication and ambiguity issues inherent with file extensions.

2. Identification

The profile for generic OPC is uniquely identified by the Canonical Identifier shown in Table G-1.

Table G-1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:gopc
Version Identifier	urn:nato:stanag:4778:profile:gopc:1:1

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

- changes to the base OPC standard
- 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 G-1.

This document deprecates the previous version identified by Version Identifier urn:nato:stanag:4778:profile:gopc:1:0.

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

3. Standards

- [1] ISO/IEC 29500-2 “Office Open XML File Formats - Part 2: Open Packaging Conventions”, at http://standards.iso.org/ittf/PubliclyAvailableStandards/c061796_ISO_IEC_29500-2_2012.zip, August 2012
- [2] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
- [3] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium

4. Notational Conventions

- 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 [3].

5. File Package

One of the common ways to package a number of files together is to use the archive file format. An archive file may contain a number of different files and an associated folder structure.

This profile adopts the Open Packaging Conventions (OPC) as defined as Part 2 of the Office Open XML specification (Reference [1]).

By adopting OPC this profile provides a structured and consistent mechanism for associating *BindingInformation* with a data object within an archive file.

This profile uses the same customXml files and relationships within the archive file as those defined in the OOXML Binding Profile, as shown in Figure G-1.

Specifically:

- A top-level relationship within the package SHALL be defined which identifies the file with which the *BindingInformation* will be associated.
- The file SHALL be held in a folder called “files”
- The *BindingInformation* SHALL be held within a file called “customXml”.
- *DataReference* elements SHALL be used to reference the files within the OPC package.
- *Data* elements SHALL NOT be used.
- *DataReference* elements used to reference the files within the OPC package will use the Pack URI scheme ‘pack’ as specified in Reference [1] Annex B.
- The authority component of the Pack URI scheme SHALL be empty that denotes the package root.
- When referring to files, or portions of files, within the OPC package, absolute URIs from the package root SHALL be used with the *DataReference* element.
- As such, a relationship is defined between the file and the *BindingInformation*.

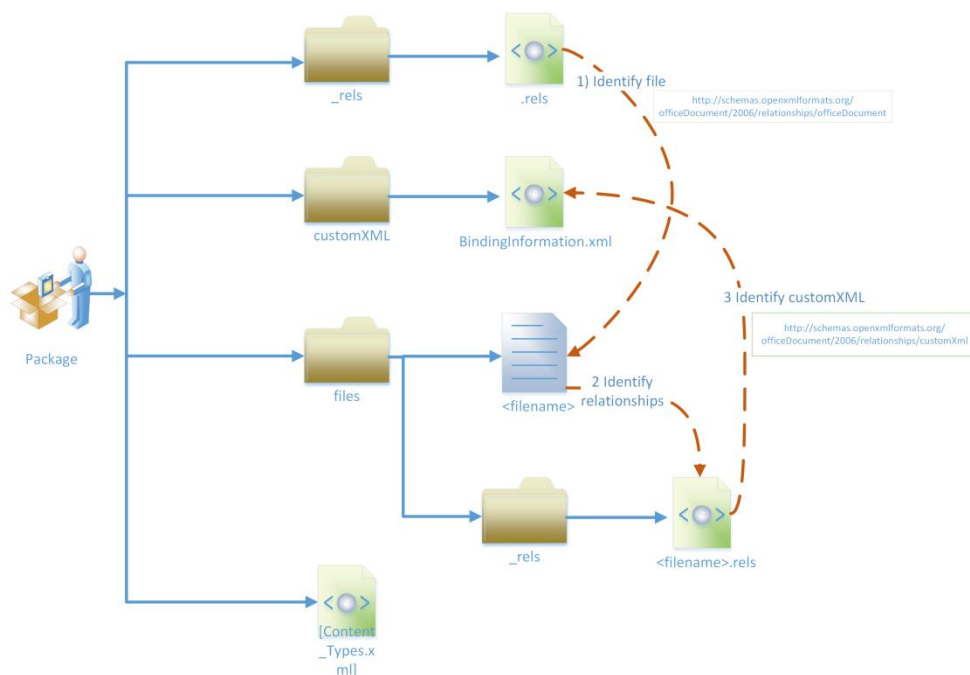


Figure G-1: OPC Structure for packaging BindingInformation with an arbitrary file

This approach allows multiple files, of different types, to be held within the same package and be bound to distinct metadata. Figure G-2 shows an example customXML file for a package containing the file “image1.jpeg”. This example uses Confidentiality Metadata Labels (Reference [2]) as example metadata.

```
<mb:BindingInformation
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
  <mb:MetadataBindingContainer>
    <mb:MetadataBinding>
      <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>
            2016-11-10T12:30:00Z
          </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
      </mb:Metadata>
      <mb:DataReference URI="pack://files/image1.jpeg" xmime:contentType="image/jpeg" />
    </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
</mb:BindingInformation>
```

Figure G-2: Example Packaged CustomXML file

6. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Annex A) SHALL be adhered to for the use cases that cryptographic bindings are required to provide a higher level of integrity protection, authenticity and non-repudiation of the binding specified in this profile.

Unless otherwise stated, all statements that apply to the Cryptographic Artefacts binding profile (Annex A) also apply to this profile for generating and validating cryptographic bindings.

ANNEX H: SIDECAR FILES BINDING PROFILE

1. Sidecar Files Introduction

If a file cannot be packaged (for example, if it is a file on a file share which needs to be accessed using the original applications), a simple naming convention to relate the BDO with the data object is proposed.

Sidecar files allow the association of metadata with a data object for which there is no profile.

This approach is well known and understood for associating data (typically metadata) with other data of a different format.

2. Identification

The profile for sidecar files is uniquely identified by the Canonical Identifier shown in Table H-1.

Table H-1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:sidecar
Version Identifier	urn:nato:stanag:4778:profile:sidecar:1:1

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

- support for specific file types
- 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 H-1.

This document deprecates the previous version identified by Version Identifier urn:nato:stanag:4778:profile:sidecar:1:0.

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

3. Standards

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

4. Notational Conventions

- 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 [3].

5. File Package

A simple naming convention is defined that allows the Binding Data Object to be maintained in a separate, but identifiable, file to the data object file, as shown in Figure H-1.

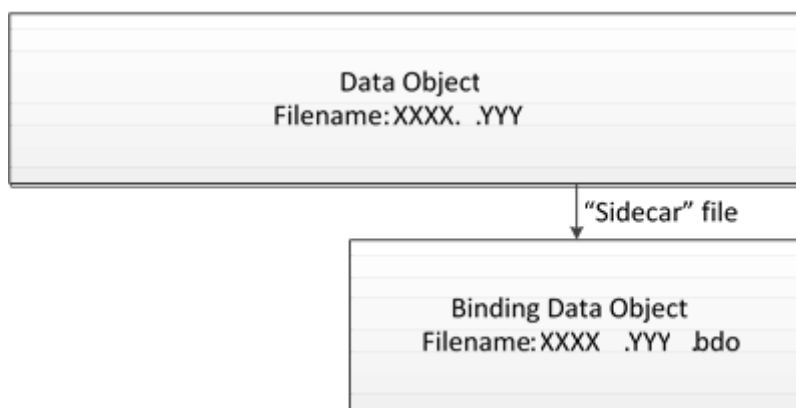


Figure H-1: BDO as a Sidecar File

The name of the Binding Data Object file SHALL be the same as the data object file, with a further “.bdo” suffix.

Values used in *DataReference* URI with the BDO SHALL use relative paths and assume that the data object resides at the same location as the BDO.

For example, distinct metadata may be associated with an image file, “image1.jpeg”, by creating a *BindingInformation* element and storing it as “image1.jpeg.bdo” in the same folder as the original file.

Figure H-2 shows an example sidecar file for “image1.jpeg”. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```
<mb:BindingInformation
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
  <mb:MetadataBindingContainer>
    <mb:MetadataBinding>
      <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>
            2016-11-10T12:30:00Z
          </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
      </mb:Metadata>
      <mb:DataReference URI="./image1.jpeg" xmime:contentType="image/jpeg" />
    </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
</mb:BindingInformation>
```

Figure H-2: Example Sidecar file

6. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Annex A) SHALL be adhered to for the use cases that cryptographic bindings are required to provide a higher level of integrity protection, authenticity and non-repudiation of the binding specified in this profile.

Unless otherwise stated, all statements that apply to the Cryptographic Artefacts binding profile (Annex A) also apply to this profile for generating and validating cryptographic bindings.

This page is left blank intentionally

ANNEX I: EXTENSIBLE METADATA PLATFORM BINDING PROFILE

1. XMP Introduction

The Extensible Metadata Platform (XMP) specifications are defined in ISO 16684-1:2012 (Reference [1]) and offer standards for the creation, processing and interchange of standardized and custom metadata for specific finite data formats.

XMP is an XML-based format modelled after the World Wide Web Consortium (W3C) Resource Description Framework (RDF) (Reference [2]) that standardizes a data model, serialization of the data model in XML, core metadata properties, definition and processing of customized metadata and a mechanism for embedding XMP information into documents, such as JPEG and PDF.

XMP offers an alternative for storing metadata in side car files whereby the XMP metadata is associated with a file format by embedding the metadata in that file format. The file formats that are supported by XMP and the locations for embedding the XMP metadata within those file formats is documented in XMP Part 3, Storage in Files (Reference [3]).

An instance of the XMP data model is called an XMP packet. An XMP packet is a set of XMP metadata properties each of which has a name and value. A value can take the form of a simple value, a structured value or an array value. This Binding Profile for XMP describes how metadata should be incorporated within an XMP packet as a simple value.

2. Identification

The profile for XMP is uniquely identified by the Canonical Identifier shown in Table I-1.

Table I-1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:xmp
Version Identifier	urn:nato:stanag:4778:profile:xmp:1:1

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

- changes to the base XMP standard
- 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 I-1.

This document deprecates the previous version identified by Version Identifier urn:nato:stanag:4778:profile:xmp:1:0.

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

3. Standards

- [1] Adobe XMP, “XMP Specification Part 1, Data Model, Serialization and Core Properties”, at <http://www.images.adobe.com/content/dam/Adobe/en/devnet/xmp/pdfs/XMP%20SDK%20Release%20cc-2016-08/XMPSpecificationPart1.pdf>, August 2016.
- [2] W3C Recommendation, “RDF Primer”, at <https://www.w3.org/TR/2004/REC-rdf-primer-20040210/>, February 2004.
- [3] Adobe XMP, “XMP Specification Part 3, Storage in Files”, at <http://www.images.adobe.com/content/dam/Adobe/en/devnet/xmp/pdfs/XMP%20SDK%20Release%20cc-2016-08/XMPSpecificationPart3.pdf>, August 2016.
- [4] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
- [5] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
- [6] W3C Recommendation, “RDF 1.1 Concepts and Abstract Syntax”, at <https://www.w3.org/TR/rdf11-concepts/>, February 2014.
- [7] W3C Recommendation, “Extensible Markup Language (XML) 1.0 (Fifth Edition)”, November 2008.

4. Notational Conventions

- 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 [3].
- Courier font indicates syntax derived from XMP (Reference [6]), RDF (Reference [2]) and XML (Reference [7]).

5. Structure

An XMP packet contains a set of XMP metadata properties, with each property having a unique name and a value. Each unique name needs to be an XML expanded name.

Values have one of three forms (Section 3 of Reference [1]):

- simple – a string of Unicode text – see Figure I-1:

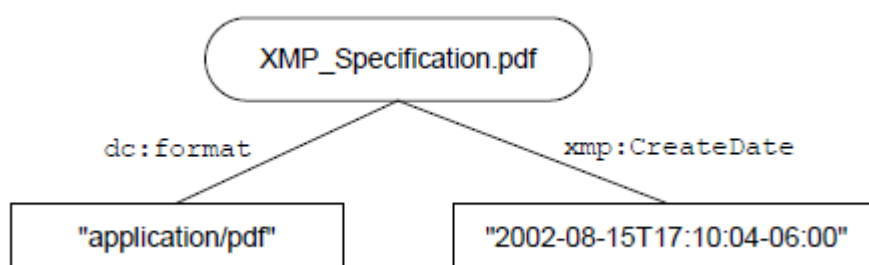


Figure I-1: Two Simple XMP Properties, `dc:format` and `xmp:CreateDate`

- structure – a container for zero or more named fields – see Figure I-2;; and

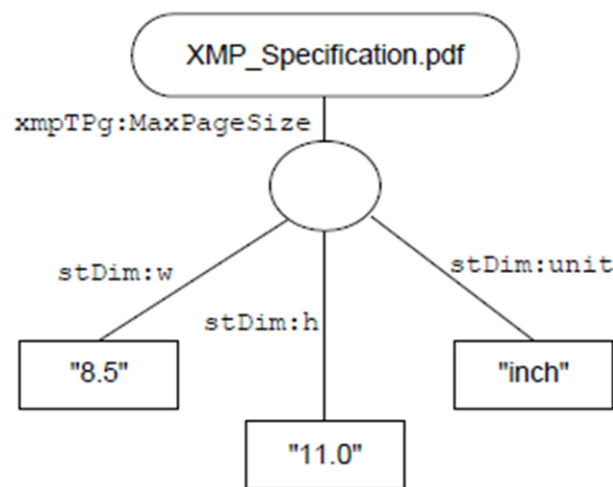


Figure I-2: An XMP Structured Property, xmpTPg:MaxPageSize containing 3 fields

- array – a container for zero or more items e.g. to support multi-valued properties – see Figure I-3:

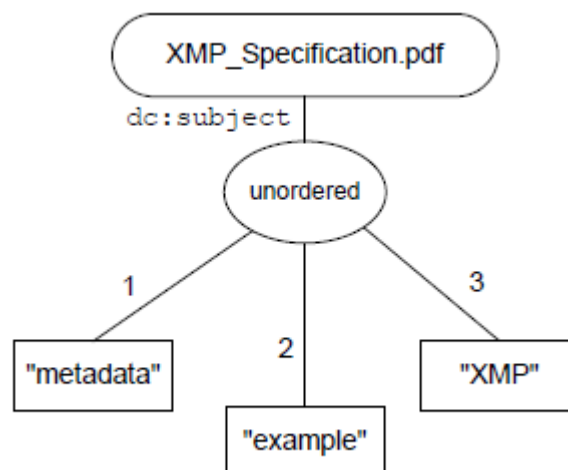


Figure I-3: An XMP Array Property, dc:subject containing 3 items

This profile defines a single metadata property with a simple form value which contains the XML markup of the BindingInformation, represented as an embedded Binding Data Object (BDO).

Specifically:

- RDF provides for XML content as a literal value. Therefore, the BindingInformation SHALL be escaped as Character Data (see Reference [6] Section 2.4) and converted to an XML literal string value compliant with Section 5.3 Reference [6]5.

⁵ Note: It is NOT REQUIRED to set the datatype to XMLLiteral.

- The BindingInformation SHALL be stored as a value within a 'bindingInformation' XML element or attribute qualified by the namespace: urn:nato:stanag:4778:bindinginformation:1:0:xmp#.
- The 'bindingInformation' XML (containing the BindingInformation) SHALL be stored as either
 - a child XML element of the rdf:Description element (canonical form – see Section 7.5 of Reference [1]); or
 - an XML attribute of rdf:Description element (equivalent form – see Section 7.9.2.2 of Reference [1])
- The serialized rdf:RDF XML element (containing the BindingInformation) is known as the XMP Binding Packet.
- The BindingInformation MUST contain at least one MetadataBinding that contains a null DataReference URI attribute value (refer to Same-Document References Section of Reference [3]) that semantically indicates a binding relationship of the metadata to the data object.
- The DataReference xmime:contentType attribute is REQUIRED when the data reference is to a non-XML entity.
- A relationship is defined between the data object and the BindingInformation by embedding the XMP Binding Packet in the data object (of a supported XMP file format).
- The supported XMP file formats are listed in Reference [3].
- Depending on the file format, the XMP Binding Packet SHALL be embedded in the data object, or held as a separate sidecar file (refer to XMP Sidecar Files Section of this profile), as specified in Reference [3].

Figure I-4 shows the structure of an XMP Binding Packet using the canonical form of the 'bindingInformation' property. This *BindingInformation* uses Confidentiality Metadata Labels (Reference [2]) as example metadata, bound to an XML entity.


```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mbxmp="urn:nato:stanag:4778:bindinginformation:1:0:xmp#" >
  <rdf:Description rdf:about="" >
    <mbxmp:bindingInformation>
      <mb:BindingInformation
        xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
        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"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
        xmlns:xmime="http://www.w3.org/2005/05/xmlmime" >
        <mb:MetadataBindingContainer>
          <mb:MetadataBinding>
            <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:ConfidentialityInformation>
                </slab:ConfidentialityInformation>
              </slab:Metadata>
            </mb:MetadataBinding>
          </mb:MetadataBindingContainer>
        </mb:BindingInformation>
      </mbxmp:bindingInformation>
    </rdf:Description>
  </rdf:RDF>

```

Figure I-4: Example XMP Binding Packet (Canonical form)

Figure I-5 shows the structure of an XMP Binding Packet using the equivalent form of the ‘bindingInformation’ property. This *BindingInformation* uses Confidentiality Metadata Labels (Reference [2]) as example metadata, bound to an XML entity.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mbxmp="urn:nato:stanag:4778:bindinginformation:1:0:xmp#" >
  <rdf:Description rdf:about=""
    xmlns:mbxmp="urn:nato:stanag:4778:bindinginformation:1:0:xmp#"
    mbxmp:bindingInformation="<mb:BindingInformation
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
      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"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:xmime="http://www.w3.org/2005/05/xmlmime"
      <mb:MetadataBindingContainer>
        <mb:MetadataBinding>
          <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:ConfidentialityInformation>
              </slab:ConfidentialityInformation>
            </mb:Metadata>
            <mb:DataReference URI="" />
          </mb:MetadataBinding>
        </mb:MetadataBindingContainer>
      </mb:BindingInformation>"
    </rdf:Description>
  </rdf:RDF>

```

Figure I-5: Example XMP Binding Packet (Equivalent form)

6. XMP Sidecar Files

If a data object file format is not supported by XMP (refer to Reference [3] to determine XMP supported file formats), XMP offers a simple naming convention to relate the XMP Binding Packet with the data object. As the XMP Binding Packet is stored separately from the data object, there is a risk that the association between the metadata and the data object may get lost. XMP-aware applications that support this profile are REQUIRED to conform with the following rules:

1. The XMP Binding Packet SHALL be written as a complete and well-formed XML document, including the leading XML declaration.
2. The base name for the XMP Binding Packet file SHALL be the same as the file to which it relates.
3. The file extension for the XMP Binding Packet file SHALL be ‘.xmp’.
4. The XMP Binding Packet file name SHALL include the base name of the file that the XMP Binding Packet relates to appended with the file extension ‘.xmp’. For example the XMP Binding Packet file name for a file named ‘example.txt’ SHALL be ‘example.txt.xmp’.
5. If a MIME type is required ‘application/rdf+xml’ SHALL be used.
6. The *BindingInformation* SHALL be represented as a detached BDO. The ‘External Storage of Media’ Section of Reference [3] states “Write external metadata as though it were embedded and then had the XMP packets extracted and catenated by a postprocessor.” However, this approach does not match the semantics for a detached BDO as described in Reference [3].
7. The *BindingInformation* MUST contain at least one *MetadataBinding*.

8. The value used in the *DataReference URI* attribute SHALL use relative paths and assume that the data object resides at the same location as the XMP Binding packet. As such, the data object file and the XMP Binding Packet file (that relates to the data object file) SHALL reside at the same location.
9. The *DataReference xmime:contentType* attribute is REQUIRED when the data reference is to a non-XML entity.

As an example, distinct metadata may be associated with an MPEG file, “example.mpg”, by creating an XMP Binding Packet containing a bindingInformation element and storing it as “example.mpg.xmp” in the same folder as the original file.

Figure I-6 shows an example XMP sidecar file for “example.mpg”. This example uses Confidentiality Metadata Labels (Reference [2]) as example metadata, bound to a non-XML entity.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mbxmp="urn:nato:stanag:4778:bindinginformation:1:0:xmp#" >
  <rdf:Description rdf:about="" >
    <mbxmp:bindingInformation>
      <mb:BindingInformation
        xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
        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"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
        xmlns:xmime="http://www.w3.org/2005/05/xmlmime" >
        <mb:MetadataBindingContainer>
          <mb:MetadataBinding>
            <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="example.mpg" >
                xmime:contentType="audio/mpeg" />
              </mb:MetadataBinding>
            </mb:MetadataBindingContainer>
          </mb:BindingInformation>
        </mbxmp:bindingInformation>
      </rdf:Description>
    </rdf:RDF>
```

Figure I-6: Example XMP Sidecar file (example.mpg.xmp)

7. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Annex A) SHALL be adhered to for the use cases that cryptographic bindings are required to provide a higher level of integrity protection, authenticity and non-repudiation of the binding specified in this profile.

Unless otherwise stated, all statements that apply to The Cryptographic Artefacts binding profile (Annex A) also apply to this profile for generating and validating cryptographic bindings.

For an embedded BDO the use of the Enveloped Binding Data Object transform element (see Annex A Transforms Section) SHALL NOT apply.

For this use case where an embedded BDO (specified in **Structure**) references a non-XML data object (indicated by the *xmime:contentType* attribute value) the XML Signature Core Generation and XML Signature Core Validation processes SHALL first exclude the embedded BDO (the *BindingInformation* element) from the digest calculation of the *Reference* element that contains the *BindingInformation* element. The *BindingInformation* element SHALL be excluded by removing the XMP Binding Packet (the serialized *rdf:RDF* XML element containing the *BindingInformation* element) from the cryptographic digest calculation.

For this use case where an embedded BDO (specified in **Structure**) references a XML data object the XML Signature Core Generation and XML Signature Core Validation processes SHALL exclude the embedded BDO (the *BindingInformation* element) from the digest calculation of the *Reference* element that contains the *BindingInformation* element. The *BindingInformation* element SHALL be excluded by removing the XMP Binding Packet (the serialized *rdf:RDF* XML element containing the *BindingInformation* element) from the cryptographic digest calculation. As such, the Enveloped Binding Data Object transform (as specified in Reference **Error! Reference source not found.**) SHALL be replaced by an Enveloped XMP Binding Packet transform.

The Enveloped XMP Binding Packet 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() = 'RDF' and
    namespace-uri() = 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'])
</XPath>
```

ANNEX J: WEB SERVICE MESSAGING PROFILE BINDING PROFILE

1. WSMP Introduction

The Web Service Messaging Profile (WSMP) defines a set of service profiles to exchange arbitrary XML-based messages. WSMP is extensible and may be used by any Community of Interest (COI). It is based on publicly available standards,

WSMP profiles a standardised messaging infrastructure able to reduce the interoperability shortfall by adopting a clear and well defined protocol and rule set. This to support the data exchange via a generic and reusable interface with the following main characteristics:

- Support of Push and Pull operations
- Usable on different communication protocols like SOAP, REST, JMS, AMQP, WEBSocket.
- Configurable for the use of different COI.

With these characteristics, WSMP is intended to be a framework for the definition of a standardised way to exchange messages.

The base of the WSMP specification are the concepts of data and metadata. Typically the relationship between the metadata and data is implicitly realized by simply including the metadata with the data in the same parent XML element.

This profile supports the requirement to explicitly bind metadata to data (or subsets thereof) whereby the data is XML-based and exchanged between service consumers and service providers using the WSMP message wrapper mechanism.

2. Standards

- [1] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
- [2] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
- [3] NCB011784-2.7-D01 v1.1, "WEB SERVICE MESSAGING PROFILE (WSMP) TECHNICAL SPECIFICATIONS (DRAFT 1.2)"

3. Identification

The profile for WSMP is uniquely identified by the Canonical Identifier shown in Table J-1.

Table J-1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:wsmp
Version Identifier	urn:nato:stanag:4778:profile:wsmp:1:1

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

- changes to the base WSMP standard
- 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 J-1.

This document deprecates the previous version identified by Version Identifier urn:nato:stanag:4778:profile:wsmpp:1:0.

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

4. Namespace Constraints

Table J-2 below summarises the XML namespaces and corresponding prefixes used throughout for the binding of metadata to SOAP data objects and portions thereof.

Table J-2: XML Namespaces and Prefixes

Prefix	Namespace
mb	urn:nato:stanag:4778:bindinginformation:1:0
wsmpp-m	urn:nato:wsmpp:1:2

5. Notational Conventions

- 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 STANAG 4778 (Reference [1]) referenced in this profile.
- Courier font indicates syntax derived from the WSMP Specification (Reference [3]) referenced in this profile.

6. WSMP Message Structure

The WSMP message that encapsulates the COI-specific data is specified in the WSMP specification (Reference [3]). A WSMP message may consist of:

1. a WSMP message wrapper with one or more WSMP data wrappers; or,
2. one or more WSMP data wrappers.

WSMP COI Profiles may specify that the data carried in the WSMP message (or subsets thereof) is bound to the metadata compliant with STANAG 4778 (Reference [1]). As such, STANAG 4778 Binding Information can be represented as follows:

1. an Embedded Binding Data Object (BDO) that binds metadata to the WSMP message wrapper `WSMPPMsg`; and/or,
2. a Detached BDO for each of the following WSMP data wrapper elements `Create`, `Read`, `Update`, `Delete` that binds metadata to the `Data` child element (or subsets thereof) of these elements.

An Embedded BDO **MUST** be present in a WSMP message that uses the WSMP message wrapper contained in the `WSMPPMsg/MetadataBinding` element that **SHALL** include the *BindingInformation* element (as a child element of the `WSMPPMsg/MetadataBinding` element).

An Embedded BDO SHALL dereference the root node of the WSMP message (WSMPMsg) by containing one null *DataReference* URI attribute value (URI=“”) and, where applicable⁶, a Transforms element containing a single Transform element that contains a Transform Algorithm attribute value of *http://www.w3.org/TR/1999/REC-xpath-19991116* and the following child XPath element:

```
<XPath>
  ancestor-or-self::*[local-name() = 'WSMPMsg' and
    namespace-uri() = 'urn:nato:wsm:1:1'
</XPath>
```

An Embedded BDO MAY contain one or more *DataReference* elements present in a *MetadataBinding* element containing a URI attribute (with optional *Transform* elements) in order to locate the data (and subsets thereof) that is contained in the WSMP Message (WSMPMsg).

For a WSMP message that consists of a WSMP message wrapper and one or more WSMP data wrapper elements (Create, Read, Update and Delete) a Detached BDO MAY be present.

For a WSMP message that consists of one or more WSMP data wrapper elements (Create, Read, Update and Delete) a Detached BDO SHALL be present.

A Detached BDO for a Create data wrapper SHALL be contained in the Create/MetadataBinding element that SHALL include the *BindingInformation* element (as a child element of the Create/MetadataBinding element).

A Detached BDO for a Read data wrapper SHALL be contained in the Read/MetadataBinding element that SHALL include the *BindingInformation* element (as a child element of the Read/MetadataBinding element).

A Detached BDO for an Update data wrapper SHALL be contained in the Update/MetadataBinding element that SHALL include the *BindingInformation* element (as a child element of the Update/MetadataBinding element).

A Detached BDO for a Delete data wrapper SHALL be contained in the Delete/MetadataBinding element that SHALL include the *BindingInformation* element (as a child element of the Delete/MetadataBinding element).

For the remainder of this normative section WSMP data wrapper elements Create, Read, Update and Delete SHALL be referred to as <data wrapper element>.

A Detached BDO SHALL contain one or more *DataReference* elements present in a *MetadataBinding* element containing a URI attribute in order to locate the data (and subsets thereof) that is contained in the WSMP message <data wrapper element>/Data element. A null *DataReference* URI attribute value (URI=“”) for a Detached BDO SHALL dereference the root node of the WSMP message <data wrapper element> element.

⁶ The WSMP Message may be contained in a higher level protocol such as SOAP. Therefore, it would be necessary to apply the transform to denote that the binding is applicable to the WSMP Message and not the SOAP message (indicated by the null *DataReference* URI attribute value (URI=“”).

For each BDO contained in a WSMP message the parent *MetadataBinding* element SHALL contain a *Dialect* attribute with the value *urn:nato:stanag:4778:bindinginformation:1:0*.

For each BDO contained in a WSMP message it is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

An example of a WSMP message (consisting of a WSMP message wrapper and an Update WSMP data wrapper element) that illustrates the binding of the data, contained in the WSMP message wrapper *WSMPMsg* and the WSMP data wrapper *WSMPMsg/Update/Data* element, to metadata is provided in Figure J-1. This example uses Confidentiality Metadata Labels (Reference [2]), referenced in this profile, as example metadata.

```
<wsmp-m:WSMPMsg
  xmlns:wsmp-m="urn:nato:wsmp:1:2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <wsmp-m:MetadataBinding Dialect="urn:nato:stanag:4778:bindinginformation:1:0">
    <mb:BindingInformation
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <mb:MetadataBindingContainer>
        <mb:MetadataBinding>
          <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>
                2016-11-20T12:30:00Z
              </slab:CreationDateTime>
            </slab:originatorConfidentialityLabel>
          </mb:MetadataBinding>
          <mb:DataReference URI=""/>
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
              <ds:XPath>
                ancestor-or-self::*[local-name()='WSMPMsg' and namespace-uri()='
urn:nato:wsmp:1:1']
              </ds:XPath>
            </ds:Transform>
          </ds:Transforms>
        </mb:MetadataBinding>
      </mb:MetadataBindingContainer>
    </mb:BindingInformation>
  </wsmp-m:MetadataBinding>
  <wsmp-m:Update>
    <wsmp-m:Data Dialect=" http://example.com/trackInformation ">
      <ns1:Track xmlns:ns1="http://example.com/trackInformation">
        ....
      </ns1:Track>
    </wsmp-m:Data>
  </wsmp-m:Update>
  <wsmp-m:MetadataBinding Dialect="urn:nato:stanag:4778:bindinginformation:1:0">
    <mb:BindingInformation
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <mb:MetadataBindingContainer>
        <mb:MetadataBinding>
          <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:originatorConfidentialityLabel>
          </mb:MetadataBinding>
          <mb:DataReference URI=""/>
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
              <ds:XPath>
                ancestor-or-self::*[local-name()='WSMPMsg' and namespace-uri()='
urn:nato:wsmp:1:1']
              </ds:XPath>
            </ds:Transform>
          </ds:Transforms>
        </mb:MetadataBinding>
      </mb:MetadataBindingContainer>
    </mb:BindingInformation>
  </wsmp-m:MetadataBinding>
  <wsmp-m:Update>
    <wsmp-m:Data Dialect=" http://example.com/trackInformation ">
      <ns1:Track xmlns:ns1="http://example.com/trackInformation">
        ....
      </ns1:Track>
    </wsmp-m:Data>
  </wsmp-m:Update>
  </wsmp-m:WSMPMsg>
```



```

    <slab:CreationDateTime>
      2016-11-20T12:30:00Z
    </slab:CreationDateTime>
    </slab:originatorConfidentialityLabel>
  </mb:Metadata>
  <mb:DataReference URI="">
    <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
        <ds:XPath>
          ancestor-or-self::*[local-name()='Data' and namespace-uri()=' urn:nato:wsmp:1:1']
        </ds:XPath>
      </ds:Transform>
    </ds:Transforms>
  </mb:DataReference>
</mb:MetadataBinding>
</mb:MetadataBindingContainer>
</mb:BindingInformation>
</wsmp-m:MetadataBinding>
</wsmp-m:Update>
</wsmp-m:WSMPMsg>

```

Figure J-1: Example WSMP Metadata Binding

7. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Annex A) SHALL be adhered to for the use cases that cryptographic bindings are required to provide a higher level of integrity protection, authenticity and non-repudiation of the binding specified in this profile.

Unless otherwise stated, all statements that apply to The Cryptographic Artefacts binding profile (Annex A) also apply to this profile for generating and validating cryptographic bindings.

This page is left blank intentionally

ANNEX K: COMMON XML ARTEFACTS BINDING PROFILE

1. Common XML Artefacts Introduction

When defining the syntax, semantics and transformation of XML-encoded data objects, a number of standard XML-encoded artefacts may typically be employed. For example, a Community of Interest (CoI) may produce a schema definition that describes the syntax of their COI-specific data objects, and a transformation that renders the data object as human-readable text.

This profile supports the requirement to bind metadata to data (or subsets thereof) whereby the data is XML-encoded in one of the following schemas:

- XML Schema – to define the syntactic structure/validation of Xml-encoded data objects (Reference [3])
- ISO Schematron – to define semantic validation (e.g. business rules) of XML-encoded data objects(Reference [4])
- XML Stylesheet – to define the transformation XML-encoded data objects (Reference [5])
- Genericcode Code List – to represent lists in a tabular form (Reference [6])
- Context/Value Association – to associate code lists with elements within XML-encoded data objects (Reference [7])
- Security Policy Information File (SPIF) – to define the value domain, equivalencies and markings instructions for a security policy used, for example, with confidentiality metadata labels (Reference [9]).

2. Identification

The profiles for XML Artefacts are uniquely identified by the Canonical Identifiers shown in Table K-1.

Table K-1: XML Artefact Profile Identifiers

XML Artefact	Type	Identifier
XML Schema	Canonical Identifier	urn:nato:stanag:4778:profile:xml:schema
	Version Identifier	urn:nato:stanag:4778:profile:xml:schema:1:0
ISO Schematron	Canonical Identifier	urn:nato:stanag:4778:profile:xml:schematron
	Version Identifier	urn:nato:stanag:4778:profile:xml:schematron:1:0
XML Stylesheet	Canonical Identifier	urn:nato:stanag:4778:profile:xml:stylesheet
	Version Identifier	urn:nato:stanag:4778:profile:xml:stylesheet:1:0
Genericcode List	Canonical Identifier	urn:nato:stanag:4778:profile:xml:codelist
	Version Identifier	urn:nato:stanag:4778:profile:xml:codelist:1:0
Context/Value Association	Canonical Identifier	urn:nato:stanag:4778:profile:xml:cva
	Version Identifier	urn:nato:stanag:4778:profile:xml:cva:1:0
Security Policy Information File	Canonical Identifier	urn:nato:stanag:4778:profile:xml:spif
	Version Identifier	urn:nato:stanag:4778:profile:xml:spif:1:0

It is recognized that these profiles may evolve during their review cycle. For example, a review might identify:

- changes to the base standards
- improvements to the existing profiles based upon operational feedback

Therefore these versions of the profiles are uniquely identified by the Version Identifier shown in Table K-1.

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

3. Standards

- [1] NATO Standardization Agency (NSA) STANAG 4774, “Confidentiality Metadata Label Syntax”, MCMSB, NATO Headquarters, Brussels, Belgium, 14 April 2016.
- [2] NATO Standardization Agency (NSA) STANAG 4778, “Metadata Binding Mechanism”, MCMSB, NATO Headquarters, Brussels, Belgium
- [3] World Wide Web Consortium (W3C) Web Standard W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures, “W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures”, M. Maloney, N. Mendelsohn, H. Thompson, D. Beech, S. Gao, M. Sperberg-McQueen, at <http://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/>, 5 April 2012.
- [4] ISO/IEC 19757-3 Second Edition 2016-01-15 – Information Technology – Document Schema Definition Languages (DSDL) – Part 3: Rules-based validation – Schematron Second Edition at http://standards.iso.org/ittf/PubliclyAvailableStandards/c055982_ISO_IEC_19757-3_2016.zip, 15 January 2016.
- [5] World Wide Web Consortium (W3C) Web Standard XSL Transformations (XSLT) Version 1.0, “XSL Transformations (XSLT) Version 1.0”, J. Clark, at <http://www.w3.org/TR/1999/REC-xslt-19991116>, 16 November 1999.
- [6] Organization for the Advancement of Structured Information Standards (OASIS) “Code List Representation (Genericcode)”, Version 1.0 , at <https://docs.oasis-open.org/codelist/cs-genericcode-1.0/doc/oasis-code-list-representation-genericcode.pdf>, 28 December 2007.
- [7] Organization for the Advancement of Structured Information Standards (OASIS) “Context/value Association using genericcode 1.0”, at <http://docs.oasis-open.org/codelist/ns/ContextValueAssociation/1.0/doc/context-value-association.pdf>, 15 April 2010.
- [8] Internet Engineering Task Force (IETF) Request for Comment 2119, “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, at <http://tools.ietf.org/html/rfc2119>, Sterling, Virginia, US, March 1997.
- [9] Security Policy Information File (SPIF) at <http://www.xmlspif.org/>.

4. Namespace Constraints

Table K-2 summarises the XML namespaces and corresponding prefixes used throughout for the binding of metadata to Common XML Artefact data objects and portions thereof.

Table K-2: XML Namespaces and Prefixes

Prefix	Namespace
mb	urn:nato:stanag:4778:bindinginformation:1:0
slab	urn:nato:stanag:4774:confidentialitymetadatalabel:1:0
xsd	http://www.w3.org/2001/XMLSchema
sch	http://purl.oclc.org/dsdl/schematron
xsl	http://www.w3.org/1999/XSL/Transform

Prefix	Namespace
gc	http://docs.oasis-open.org/codelist/ns/genericcode/1.0/
cva	http://docs.oasis-open.org/codelist/ns/ContextValueAssociation/1.0/
spif	http://www.xmlspif.org/spif

5. Notational Conventions

- 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 Reference [8].
- Words in *italics* indicate terms defined in Appendix 1 of Reference [2].
- Courier font indicates syntax derived from the Specifications referenced in this Profile.

6. XML Schema Structure

The XML Schema contains an `xsd:annotation` element which allows for both human readable and machine-processible, inline documentation to be provided for any element within the schema. The `xsd:annotation` element has a child element, `xsd:appinfo`, which allows any well-formed XML content to be included within the annotation. The Binding Information can thus be included within the `xsd:appinfo` element.

As such, the Binding Information SHALL be represented as an Embedded BDO.

A BDO SHALL be embedded within the XML Schema as a child `mb:BindingInformation` of the `xsd:appinfo` element of the `xsd:annotation` element(s) of the top-level `xsd:schema` element. (XPath: `/xsd:schema/xsd:annotation/xsd:appinfo/mb:BindingInformation`).

A BDO SHALL NOT be embedded in any other location within the XML Schema.

Multiple BDOs MAY be embedded as child `mb:BindingInformation` elements of a single `xsd:appinfo` element.

Multiple BDOs MAY be embedded as child `mb:BindingInformation` elements of distinct `xsd:appinfo` elements.

It is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

One or more *DataReference* elements SHALL be present in a *MetadataBinding* element containing a *URI* attribute in order to locate the data (and subsets thereof) that is contained in the `xsd:schema` top-level element.

An example of an BDO embedded in a XML Schema that illustrates the binding of the data, contained in the parent `xsd:schema` element, to metadata is provided in Figure K-1. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```
<xsd:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://example.com/simpleSchema"
  xmlns:tns="http://example.com/simpleSchema" version="1.0">
  <xsd:annotation>
    <xsd:appinfo>
      <mb:BindingInformation xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
```

```

<mb:MetadataBindingContainer>
  <mb:MetadataBinding>
    <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>2016-11-20T12:30:00Z</slab:CreationDateTime>
      </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI=""/>
  </mb:MetadataBinding>
</mb:MetadataBindingContainer>
</mb:BindingInformation>
</xsd:appinfo>
</xsd:annotation>
<xsd:simpleType name="exampleType">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

Figure K-1: Example XML Schema Metadata Binding.

7. Schematron Structure

The Schematron (<https://www.w3.org/2007/schema-for-xslt20> xsd) allows any element from a different schema to be included within the top-level element of the schematron. The Binding Information can thus be included within the `sch:schema` element.

As such, the Binding Information SHALL be represented as an Embedded BDO.

A BDO SHALL be embedded within the Schematron as a child `mb:BindingInformation` element of the top-level `sch:schema` element. (XPath: `/sch:schema/mb:BindingInformation`).

A BDO SHALL NOT be embedded in any other location within the Schematron.

Multiple BDOs MAY be embedded as child `mb:BindingInformation` elements of the top level `sch:schema` element.

It is RECOMMENDED that the `mb:BindingInformation` elements be placed at the start of the stylesheet, as the first child element of the `sch:schema` element.

It is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

One or more *DataReference* elements SHALL be present in a *MetadataBinding* element containing a *URI* attribute in order to locate the data (and subsets thereof) that is contained in the `sch:schema` top-level element.

An example of an BDO embedded in an Schematron that illustrates the binding of the data, contained in the parent `sch:schema` element, to metadata is provided in Figure K-2. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron">
  <mb:BindingInformation xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
    <mb:MetadataBindingContainer>
      <mb:MetadataBinding>
        <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>2016-11-20T12:30:00Z</slab:CreationDateTime>
          </slab:originatorConfidentialityLabel>
        </mb:Metadata>
        <mb:DataReference URI=""/>
      </mb:MetadataBinding>
    </mb:MetadataBindingContainer>
  </mb:BindingInformation>
  <sch:title>Example Schematron</sch:title>
  <sch:rule context="example">
    <sch:assert test="@example">Example</sch:assert>
  </sch:rule>
</sch:schema>
```

Figure K-2: Example XML Schematron Metadata Binding

8. XML Stylesheet Structure

The XML Stylesheet (<https://www.w3.org/2007/schema-for-xslt20> xsd) allows any element from a different schema to be included within the top-level element of the XML stylesheet, after any `xsl:import` elements. The Binding Information can thus be included within the `xsl:stylesheet` element.

As such, the Binding Information SHALL be represented as an Embedded BDO.

A BDO SHALL be embedded within the XML Stylesheet as a child `mb:BindingInformation` element of the top-level `xsl:stylesheet` element. (XPath: `/xsl:stylesheet/mb:BindingInformation`).

A BDO SHALL NOT be embedded in any other location within the XML Stylesheet.

Multiple BDOs MAY be embedded as child `mb:BindingInformation` elements of the top level `xsl:stylesheet` element.

It is RECOMMENDED that the `mb:BindingInformation` elements be placed at the start of the stylesheet, immediately after the `xsl:import` elements, if present.

It is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

One or more *DataReference* elements SHALL be present in a *MetadataBinding* element containing a *URI* attribute in order to locate the data (and subsets thereof) that is contained in the `xsl:stylesheet` top-level element.

An example of an BDO embedded in an XML Stylesheet that illustrates the binding of the data, contained in the parent `xsl:stylesheet` element, to metadata is provided in Figure K-3. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:import href="example.xml"/>
  <mb:BindingInformation xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
    <mb:MetadataBindingContainer>
      <mb:MetadataBinding>
        <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>2016-11-20T12:30:00Z</slab:CreationDateTime>
          </slab:originatorConfidentialityLabel>
        </mb:Metadata>
        <mb:DataReference URI=""/>
      </mb:MetadataBinding>
    </mb:MetadataBindingContainer>
  </mb:BindingInformation>
  <xsl:output method="text"/>
  <xsl:template match="/">
    <xsl:text>Example</xsl:text>
  </xsl:template>
</xsl:stylesheet>
```

Figure K-3: Example XML Stylesheet Metadata Binding

9. Genericcode Code List Structure

The Genericcode Code List (<https://docs.oasis-open.org/codelist/cs-genericcode-1.0/xsd/genericcode.xsd>) contains an *Annotation*⁷ element which allows for both human readable and machine-processable, inline, documentation to be provided for any element within the schema. The *Annotation* element has a child element, *AppInfo*, which allows any well-formed XML content to be included within the annotation. The Binding Information can thus be included within the *AppInfo* element.

As such, the Binding Information SHALL be represented as an Embedded BDO.

A BDO SHALL be embedded within the Genericcode CodeList as a child `mb:BindingInformation` of the `AppInfo` element of the *Annotation* element(s) of the top-level `gc:CodeList` element. (XPath: `/gc:CodeList /Annotation/AppInfo/mb:BindingInformation`).

⁷ The `gc:CodeList` child elements have no namespace.

A BDO SHALL NOT be embedded in any other location within the Genericcode Code List.

Multiple BDOs MAY be embedded as child `mb:BindingInformation` elements of a single `AppInfo` element.

Multiple BDOs MAY be embedded as child `mb:BindingInformation` elements of distinct `AppInfo` elements.

It is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

One or more *DataReference* elements SHALL be present in a *MetadataBinding* element containing a *URI* attribute in order to locate the data (and subsets thereof) that is contained in the `gc:CodeList` top-level element.

An example of an BDO embedded in a Genericcode CodeList that illustrates the binding of the data, contained in the parent `gc:CodeList` element, to metadata is provided in Figure K-4. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```
<gc:CodeList xmlns:gc=" http://docs.oasis-open.org/codelist/ns/genericcode/1.0/"
  <Annotation>
    <AppInfo>
      <mb:BindingInformation xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
        <mb:MetadataBindingContainer>
          <mb:MetadataBinding>
            <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>2016-11-20T12:30:00Z</slab:CreationDateTime>
              </slab:originatorConfidentialityLabel>
            </mb:Metadata>
            <mb:DataReference URI=""/>
          </mb:MetadataBinding>
        </mb:MetadataBindingContainer>
      </mb:BindingInformation>
    </AppInfo>
  </Annotation>
  <Identification>
    <ShortName>example</ShortName>
  </Identification>
  <ColumnSet>
    <Column id="id">
      <ShortName>ID</ShortName>
      <Data Type="xsd:string"/>
    </Column>
    <Column id="price">
      <ShortName>Price</ShortName>
      <Data Type="xsd:string"/>
    </Column>
  </ColumnSet>
  <SimpleCodeList>
    <Row>
      <Value ColumnRef="id"><SimpleValue>1</SimpleValue></Value>
      <Value ColumnRef="price"><SimpleValue>100</SimpleValue></Value>
```

```

    <Row>
  </SimpleCodeList>
</gc:CodeList>

```

Figure K-4: Example XML Genericode Metadata Binding

10. Context/Value Association Structure

The Context/Value Association (<http://docs.oasis-open.org/codelist/cs01-ContextValueAssociation-1.0/xsd/ContextValueAssociation.xsd>) contains an `cva:Annotation` element which allows for both human readable and machine-processible, inline, documentation to be provided for any element within the schema. The `cva:Annotation` element has a child element, `cva:AppInfo`, which allows any well-formed XML content to be included within the annotation. The Binding Information can thus be included within the `cva:AppInfo` element.

As such, the Binding Information SHALL be represented as an Embedded BDO.

A BDO SHALL be embedded within the Context/Value Association as a child `mb:BindingInformation` of the `cva:AppInfo` element of the `cva:Annotation` element(s) of the top-level `cva:ContextValueAssociation` element. (XPath: `/cva:ContextValueAssociation /cva:Annotation/cva:AppInfo/mb:BindingInformation`).

A BDO SHALL NOT be embedded in any other location within the Context/Value Association.

Multiple BDOs MAY be embedded as child `mb:BindingInformation` elements of a single `cva:AppInfo` element.

Multiple BDOs MAY be embedded as child `mb:BindingInformation` elements of distinct `cva:AppInfo` elements.

It is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

One or more *DataReference* elements SHALL be present in a *MetadataBinding* element containing a *URI* attribute in order to locate the data (and subsets thereof) that is contained in the `cva:ContextValueAssociation` top-level element.

An example of an BDO embedded in a Context/Value Association that illustrates the binding of the data, contained in the parent `cva:ContextValueAssociation` element, to metadata is provided in Figure K-5. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```

<cva:ContextValueAssociation
  xmlns:cva="http://docs.oasis-open.org/codelist/ns/ContextValueAssociation/1.0/"
  name="exampleCVA" version="1.0">
  <cva:Annotation>
    <cva:AppInfo>
      <mb:BindingInformation xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
        <mb:MetadataBindingContainer>
          <mb:MetadataBinding>
            <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>2016-11-20T12:30:00Z</slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI=""/>
</mb:MetadataBinding>
</mb:MetadataBindingContainer>
</mb:BindingInformation>
</cva:AppInfo>
</cva:Annotation>
<cva:Title>Example CVA</cva:Title>
    <cva:ValueLists>
        <cva:ValueList xml:id="exampleCodes-v1" uri="CodeLists/exampleCode-v1.gc"/>
    </cva:ValueLists>
<cva:Contexts>
    <cva:Context address="example" values="exampleCode-v1" />
</cva:Contexts>
</cva:ContextValueAssociation>

```

Figure K-5: Example XML Context/Value Association Metadata Binding

11. Security Policy Information File Structure

The Security Policy Information File (<http://www.xmlspif.org/schema/xmlspif.xsd>) contains an `spif:extensions` element which allows for arbitrary extensions to be included within the SPIF.

The Binding Information can thus be included within the `spif:extensions` element.

As such, the Binding Information SHALL be represented as an Embedded BDO.

A BDO SHALL be embedded within the SPIF as a child `mb:BindingInformation` of the `spif:extensions` element of the top-level `spif:SPIF` element. (XPath: `/spif:SPIF/spif:extensions/mb:BindingInformation`).

A BDO SHALL NOT be embedded in any other location within the SPIF.

Multiple BDOs MAY be embedded as child `mb:BindingInformation` elements of a single `spif:extensions` element.

It is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

One or more *DataReference* elements SHALL be present in a *MetadataBinding* element containing a *URI* attribute in order to locate the data (and subsets thereof) that is contained in the `spif:SPIF` top-level element.

An example of an BDO embedded in a SPIF that illustrates the binding of the data, contained in the parent `spif:SPIF` element, to metadata is provided in Figure K-6. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

```

<spif:SPIF xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:spif="http://www.xmlspif.org/spif"
  schemaVersion="1.0" version="1" creationDate="20170330150423Z"

```

```

originatorDN="CN=SPIF ADMIN,O=SMHS Ltd,C=GB"
keyIdentifier="6AA4BA9F66BFCD44"
privilegeId="1.3.6.1.4.1.31778.110.110"
rbacId="1.3.6.1.4.1.31778.110.110">
<spif:securityPolicyId name="CWIX17" id="1.3.6.1.4.1.31778.102.17" />
<spif:securityClassifications>
  <spif:securityClassification name="UNCLASSIFIED" lacv="1" hierarchy="1">
    <spif:markingData xml:lang="fr" phrase="SANS CLASSIFICATION">
      <spif:code>documentStart</spif:code>
    </spif:markingData>
  </spif:securityClassification>
  <spif:securityClassification name="RESTRICTED" lacv="2" hierarchy="2">
    <spif:markingData xml:lang="fr" phrase="DIFFUSION RESTREINTE">
      <spif:code>documentStart</spif:code>
    </spif:markingData>
  </spif:securityClassification>
  <spif:securityClassification name="CONFIDENTIAL" lacv="3" hierarchy="3">
    <spif:markingData xml:lang="fr" phrase="CONFIDENTIEL">
      <spif:code>documentStart</spif:code>
    </spif:markingData>
  </spif:securityClassification>
  <spif:securityClassification name="SECRET" lacv="4" hierarchy="4">
    <spif:markingData xml:lang="fr" phrase="SECRET">
      <spif:code>documentStart</spif:code>
    </spif:markingData>
  </spif:securityClassification>
</spif:securityClassifications>
<spif:extensions>
  <BindingInformation xmlns="urn:nato:stanag:4778:bindinginformation:1:0">
    <MetadataBindingContainer>
      <MetadataBinding xml:id="id-4ec8e07f-2336-4ee0-af34-1e7f15f946ea">
        <Metadata xml:id="id-d3e4fa3b-4318-4a65-9eba-53341c3fb92d">
          <slab:originatorConfidentialityLabel
            xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
            xmlns:slab-ext="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0:ext">
            <slab:ConfidentialityInformation>
              <slab:PolicyIdentifier>NATO</slab:PolicyIdentifier>
              <slab:Classification>UNCLASSIFIED</slab:Classification>
              <slab-ext:Marking xml:lang="en">NATO UNCLASSIFIED</slab-ext:Marking>
            </slab:ConfidentialityInformation>
            <slab:CreationDateTime>2015-09-30T12:30:00Z</slab:CreationDateTime>
          </slab:originatorConfidentialityLabel>
        </Metadata>
        <DataReference URI="" />
      </MetadataBinding>
    </MetadataBindingContainer>
  </BindingInformation>
</spif:extensions>
</spif:SPIF>

```

Figure K-6: Example XML SPIF Metadata Binding

12. Cryptographic Artefacts Profile

The Cryptographic Artefacts binding profile (Annex A) SHALL be adhered to for the use cases that cryptographic bindings are required to provide a higher level of integrity protection, authenticity and non-repudiation of the binding specified in this profile.

Unless otherwise stated, all statements that apply to The Cryptographic Artefacts binding profile (Annex A) also apply to this profile for generating and validating cryptographic bindings.

